Stop wasting Time On Determined LOL match

Runze Wang

March 21, 2024

Abstract

This project is designed to predict the result of a League of Legend Esport match with the data from the first 15 miniutes into the game.

Github: DSC148 LOL 15minPred Project
https://github.com/RunzeWang0728/DSC148_LOL_15minPred

1 Introduction

League of Legends (LOL) have one of the most competitive Esport match in the world. However, audience on social media such as Reddit and HuPu (one of the biggest platform in China for rating) have realize that the early advantages in LOL matches could easily be snowballed to a victory. Therefore, most of the audience would only watch the first half of the match and the game would be determined as expected. Therefore, I decide to carry out an analysis to check whether a team will win in the end based on the data performance of a team in the early stages of the game in 2023 professional League of Legends competition

2 Dataset

2.1 The dataset

The dataset I am using is from Oracle Elixir. (https://oracleselixir.com/tools/downloads) Oracle Elixir contains the advanced League of Legends Esports statistics. The dataset for 2023 Esport match contains various valuable information such as first dragon taken and gold difference at 15 miniutes mark. The data snippet is shown as below:

	gameid	datacompleteness	url	league	year	split	playoffs	date	game	patch	...	opp_csat15
0	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	131.0
1	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	117.0
2	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	162.0
3	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	122.0
4	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	3.0
...	...	...	...	...	...	...	...	...	...	...	...	...
125899	10659-10659_game_4	partial	https://lpl.qq.com/es/stats.shtml?bmid=10659	DCup	2023	NaN	0	2023-12-31 11:48:06	4	13.24	...	NaN
125900	10659-10659_game_4	partial	https://lpl.qq.com/es/stats.shtml?bmid=10659	DCup	2023	NaN	0	2023-12-31 11:48:06	4	13.24	...	NaN
125901	10659-10659_game_4	partial	https://lpl.qq.com/es/stats.shtml?bmid=10659	DCup	2023	NaN	0	2023-12-31 11:48:06	4	13.24	...	NaN
125902	10659-10659_game_4	partial	https://lpl.qq.com/es/stats.shtml?bmid=10659	DCup	2023	NaN	0	2023-12-31 11:48:06	4	13.24	...	NaN
125903	10659-10659_game_4	partial	https://lpl.qq.com/es/stats.shtml?bmid=10659	DCup	2023	NaN	0	2023-12-31 11:48:06	4	13.24	...	NaN

25904 rows × 131 columns

As shown in the snippet, the Dataframe contains 25904 rows and 131 columns. It has valuable features, but we need to clean the dataframe for our analysis.

2.2 Data Cleanning

To achieve the data for our analysis, we first need to get the complete data without missing values. Firstly, the raw dataset has a column called "datacompleness". This is useful as the row with "complete" has the complete data.
Then, we find out the dataset contain the value for team performance or each position player's performance. Therefore, I choose to limit the analysis on only team performance. The intermediate dataframe is shown as below:

	gameid	datacompleteness	url	league	year	split	playoffs	date	game	patch	...	opp_csat15	golddiffat15	xpdiffat15	csdi
0	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	131.0	322.0	263.0	
1	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	117.0	-357.0	-1323.0	
2	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	162.0	-479.0	-324.0	
3	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	122.0	200.0	292.0	
4	ESPORTSTMNT06_2753012	complete	NaN	LFL2	2023	Spring	0	2023-01-10 17:07:16	1	13.01	...	3.0	-216.0	-579.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
125407	ESPORTSTMNT01_3438678	complete	NaN	NEXO	2024	Split 1	0	2023-11-20 19:59:21	2	13.22	...	135.0	131.0	66.0	
125408	ESPORTSTMNT01_3438678	complete	NaN	NEXO	2024	Split 1	0	2023-11-20 19:59:21	2	13.22	...	129.0	-381.0	1106.0	
125409	ESPORTSTMNT01_3438678	complete	NaN	NEXO	2024	Split 1	0	2023-11-20 19:59:21	2	13.22	...	27.0	-142.0	-769.0	
125410	ESPORTSTMNT01_3438678	complete	NaN	NEXO	2024	Split 1	0	2023-11-20 19:59:21	2	13.22	...	515.0	-660.0	-1568.0	
125411	ESPORTSTMNT01_3438678	complete	NaN	NEXO	2024	Split 1	0	2023-11-20 19:59:21	2	13.22	...	481.0	660.0	1568.0	

105924 rows × 131 columns

To further find the useful features for our prediction, we need to study each column. After going over all the columns, I pick the following columns for our analysis:
'firstblood': float value to determine whether the team obtain the first kill in the game. 1.0 represents true and 0.0 represents false.

'firstdragon': float value to determine whether the team obtain the first dragon in the game. 1.0 represents true and 0.0 represents false.

'firsttower': float value to determine whether the team destroy the first tower in the game. 1.0 represents true and 0.0 represents false.

'golddiffat15': float value for the gold difference with the opposing team.

'csdiffat15': float value for the cs(creep score) difference with the opposing team.

'xpdiffat15': float value for the experience difference with the opposing team.

'result': int value to determine if the team wins or loses. 1 represents true and 0 represents false.

In addition, I create the following new columns:

"killdiffat15": float value represents the kill difference created by using kills at 15 mins minus kills by the opposing team.

"assistdiffat15": float value represents the assist difference created by using assist at 15 mins minus assist by the opposing team

"deathdiffat15": float value represent the death difference created by using death at 15 mins minus assist by the opposing team

	firstblood	firstdragon	firsttower	result	golddiffat15	csdiffat15	xpdiffat15	killdiffat15	assistdiffat15	deathdiffat15
10	0.0	0.0	1.0	1	-530.0	-37.0	-1671.0	-1.0	-1.0	1.0
11	1.0	1.0	0.0	0	530.0	37.0	1671.0	1.0	1.0	-1.0
22	0.0	0.0	0.0	0	673.0	-34.0	530.0	1.0	2.0	-1.0
23	1.0	1.0	1.0	1	-673.0	34.0	-530.0	-1.0	-2.0	1.0
34	0.0	0.0	0.0	1	-1901.0	58.0	-763.0	-2.0	-5.0	2.0
...	...	...	...	...	...	...	...	...	...	...
125387	0.0	1.0	0.0	0	31.0	19.0	1299.0	3.0	10.0	-2.0
125398	1.0	1.0	1.0	1	-75.0	54.0	545.0	-2.0	-5.0	2.0
125399	0.0	0.0	0.0	0	75.0	-54.0	-545.0	2.0	5.0	-2.0
125410	1.0	0.0	0.0	1	-660.0	-34.0	-1568.0	2.0	7.0	-2.0
125411	0.0	1.0	1.0	0	660.0	34.0	1568.0	-2.0	-7.0	2.0

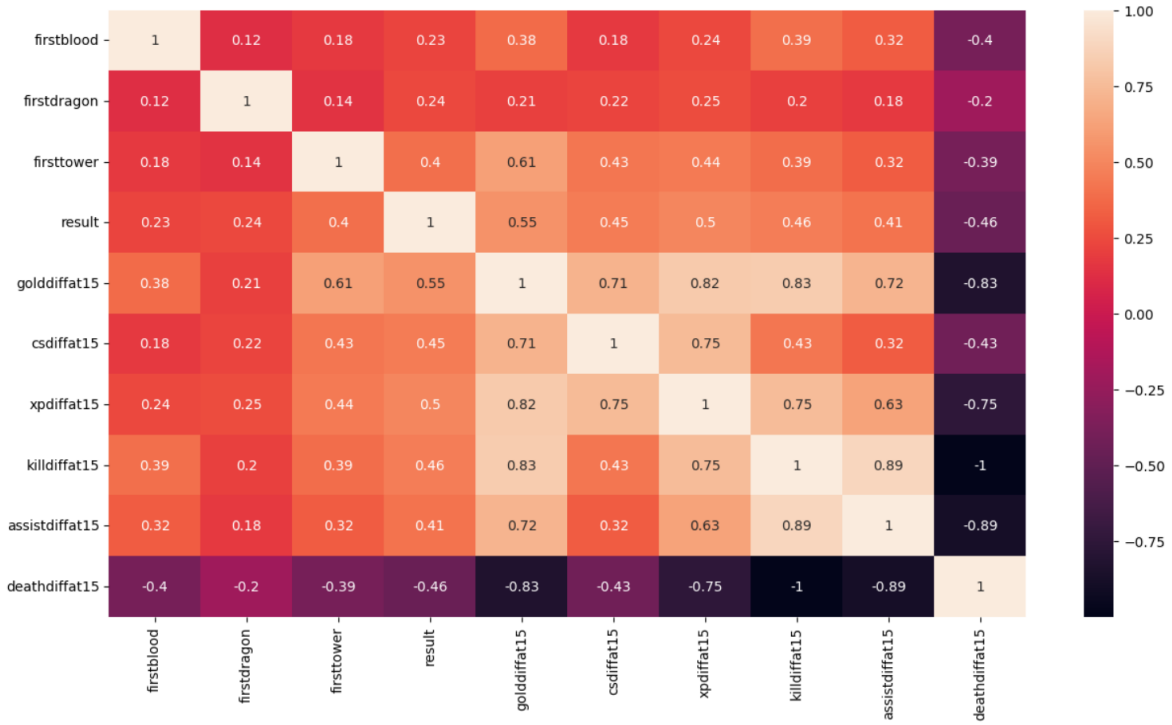
17654 rows × 10 columns

2.3 Data Statistics

Below is the data statistics:

	firstblood	firstdragon	firsttower	result	golddiffat15	csdiffat15	xpdiffat15	killdiffat15	assistdiffat15	deathdiffat15
count	17654.000000	17654.000000	17654.000000	17654.000000	17654.000000	17654.0000	17654.000000	17654.00000	17654.000000	17654.000000
mean	0.499264	0.499830	0.499943	0.499943	0.000000	0.0000	0.000000	0.00000	0.000000	0.000000
std	0.500014	0.500014	0.500014	0.500014	3075.916366	39.8683	1907.271833	3.58182	6.702235	3.580855
min	0.000000	0.000000	0.000000	0.000000	-17056.000000	-202.0000	-9580.000000	-20.00000	-33.000000	-20.000000
25%	0.000000	0.000000	0.000000	0.000000	-1923.750000	-26.0000	-1190.000000	-2.00000	-4.000000	-2.000000
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.0000	0.000000	0.00000	0.000000	0.000000
75%	1.000000	1.000000	1.000000	1.000000	1923.750000	26.0000	1190.000000	2.00000	4.000000	2.000000
max	1.000000	1.000000	1.000000	1.000000	17056.000000	202.0000	9580.000000	20.00000	33.000000	20.000000

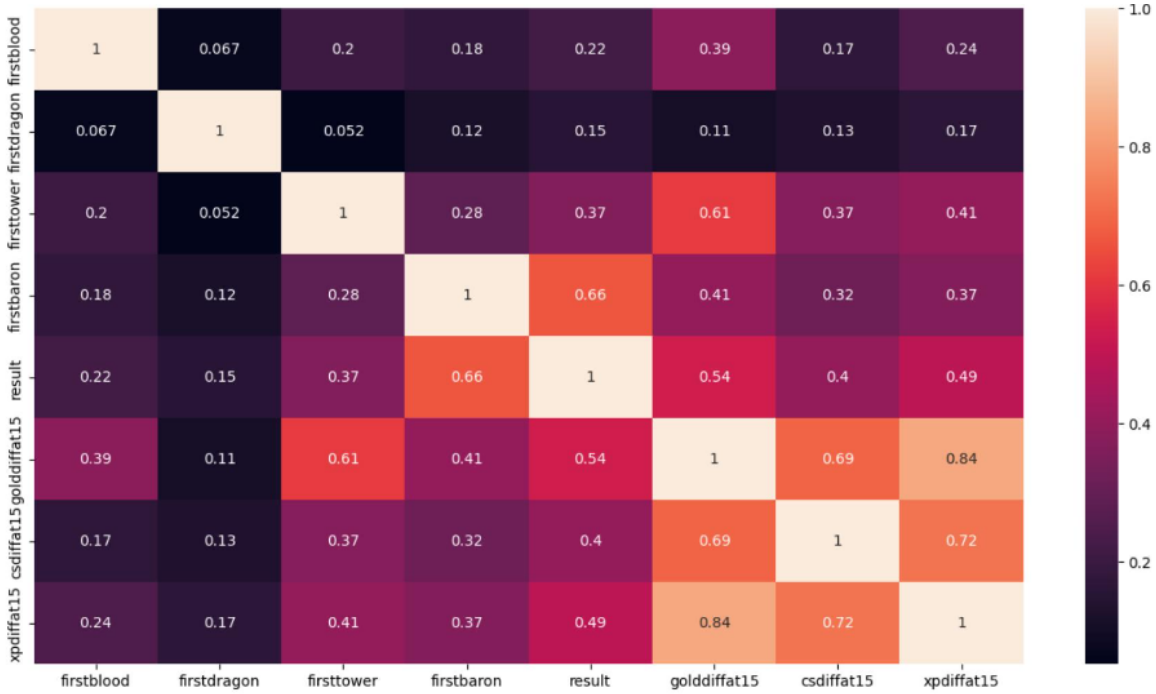
we will perform a correlation matrix to check the correlation between the features and the results:



Interesting findings: based on the correlation matrix, I find out that golddiffat15 is the most significant factor for a winning game and the second is xp diff. The deathdiffat15 has negative correlation as we expected. The more death the team has, the harder to win.

Another interesting finding is the change of the current meta for the strategy for teams.

I perform another similar process to 2022 Esport data and build the correlation map as followed:



From the comparison between 2022 and 2023 data, we could tell some change of meta by the developer:

1, Developers have a more emphasis on the first dragon. In the 2022 data, first dragon has only 0.15 correlation with the result of the game. However, in 2023, it moves to 0.22. Therefore, team could start to change the strategy as the importance of dragon increases.

2, First blood becomes important. the correlation with the result of the game increases from 0.22 to 0.38. Therefore, players would become more aggressive during the early stages in order to get the first blood in the game. This is already shown in the matches. In the 2022 matches, jungle would prefer go to bottom lane first for defensive purposes. In the 2023 matches, Jungle prefer to gank Top first to seize the first pick (first blood).

2.4 Predictive Task

Prediction: Whether a team will win in the end based on the data performance of a team in the early stages (15 miniutes) of the game in 2023 professional League of Legends competition.

3 Baseline Model

I firstly build a common column transformer in order to handle the categorical value. Then I decide to test 7 models to check which one is more suitable. The models I use are:

Logistic Regression, KNN, Decision Tree, Random Forest, SVC, Ada Boosting, Gradient Boosting

These models can be great baseline models because they cover from the simple model to some complex model. With these models, we can tell whether the data has simple correlation such as linearly correlation or some intricate correlations. The accuracy and F1-score is shown as followed:

	Method	Accuracy
0	Logistic Regression	0.743274
1	Decision Tree	0.707590
2	KNN	0.701643
3	Random Forest	0.736477
4	SVC	0.738035
5	AdaBoost	0.735911
6	Gradient Boosting	0.737751

```
logreg F1 score: 0.7441
tree F1 score: 0.7105
knn F1 score: 0.7067
rf F1 score: 0.7385
svc F1 score: 0.7398
ada F1 score: 0.7344
grad F1 score: 0.7394
```

From the statistics above, we find out that Logistic Regression performs the best as the baseline. It is quite surprising at beginning, but it is reasonable because the features and the result is approximately linear since each of them has significance on the prediction.

4 Final Model

For the final mode, I decide to use a Random Forest Classifier. Lol match outcomes are influenced by complex iteractions between various game aspects. Random Forest can be a great model for final model because it has already captured some non-linear relationships without other tuning.

There are a lot of unsuccessful tries in building the final model. I begin by using Logistic Regression, but it shows the linear relationships so it provide little insights and less effective to show the correlation. Therefore, with better tuning, I believe that Random Forest would be the best models.

Based on result from the baseline models, I decide to improve the Random Forest model by using MaxAbsScaler and Normalizer. Then perform a GridSearch using the pipeline containing random forest model. I tune the parameter of Grids and find out the best estimators are 200 and the max depth is 20.

```
pipeline = Pipeline([
    ('transformer', transformer),
    ('MaxAbsScale', MaxAbsScaler()),
    ('normalize', Normalizer()),
    ('model', RandomForestClassifier(random_state=42))
])


param_grid = {
    'model__n_estimators': [150, 200, 250],
    'model__max_depth': [None, 20, 120, 150],
}

grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='accuracy')

grid_search.fit(X_train, y_train)

print("Best parameters:", grid_search.best_params_)
print("Best score:", grid_search.best_score_)
```

```
Best parameters: {'model__max_depth': 20, 'model__n_estimators': 200}
Best score: 0.7466960425058411
```

There are a lot of unsuccessful tries in building the final model. I begin by using Logistic Regres-

sion, but it shows the linear relationships so it provide little insights and less effective to show the correlation. Therefore, with better tuning, I believe that Random Forest would be the best models.

Scalability: it takes ages to run the final models because of grid search and random forest classifier. Therefore, I decide to find the best n-estimators and use it for the interaction.

Overfirtting: Random Forest is generally overfitting. Therefore, Grid search would be the best option to find the maximum depth so that the overfitting could be limited.

We could see that the final model produce 0.747 accuracy which is better than the best performing baseline model (0.744)

5 Analysis/Literature

Prediction in Esport is a large industry. This study focuses on identifying key factors that contribute to a success and develop models that can accurate predict match outcomes based on the factors.

Usage: The dataset is valuable as it provides various factors such as kills, deaths, gold earned, and objectives. it offers performance analysis to understand how different strategies and player performances influence the game outcome.

For example, we could tell that gold differences matters the most determining the outcome of the game and the first dragon is the least effective in determining the outcome. Therefore, the teams could set up strategies around the useful information from this study: First dragon could give to the other team, and we focus on farming golds. Moreover, teams could understand the current meta better. As the comparison of correlation map above, we could see the importance of the first blood. This could make the players more aggressive and offer more entertainments for the audiences.

Novelty of the Task: Prediction in Esport Industry is not a brand0new task, however, my approach to use only the early stages match data is a fresh perspective. Early game dynamics is different from the full-match data. Focusing on this phase could show the critical factors to the victory and whether it is true that audience only needs to see the first 15 minutes of the game and the outcome of the game is already determined.

Method by the others: There are a lot of researches on predicting the outcome of the game. As I show in my baseline model, logistic regression performs the best, so the majority of the prediction use such a method. However, people with better ability prefer to run neural networks for a more precise outcome (often yield 80+ accuracy). For the pipeline method, the majority of us chooses Random Forests and Gradient Boosting for interpret ability and complexity.

During the middle of the game, the official prediction will offer by given the past data collected of the team. This is because each team has different play style: some focus on early game ganks and some focus on middle game team fights. This offers better prediction accuracy because of the vast data and specific individual data on the team. For example, IG is famous for early aggressiveness, and they could end the game by snowballing the early advantages. However, without early advantages, IG has a significant decrease in the win rate. Therefore, offocial prediction could offer more insights than my general predictions.

One thing to point out is that the method we choose shift significantly. The focus on the prediction changes. Some focus on the importance of the sides (Blue and Red) and some focus on the individual player's performance. My study focuses on the team performance and the gap (i.e. difference between golds)of the data.

consistency with the others: Yes, my study is consistent with the other studies as the early advantages is significantly important in winning the game. This is widely known that professional players could easily control the advantages and snowball to the end of the game.

Results: My final model successfully outperform the best baseline models. This is because random forest classifier with tuning could handle more complex data relationship. The gap is not significant because the features are pre-handled so it fits well with the baseline models causing the space for improvement is limited.

One interesting case is EDG vs BLG. The relative data are: 'firstblood': 1.0, 'firstdragon': 1.0, 'firsttower': 1.0, 'golddiffat15':6042.0, 'csdiffat15':153.0, 'xpdiffat15': 2508.0, 'result':0.0

This is iconic because the features are significantly high, and the result would be 1.0 with 100 percent. However, BLG had successfully turned the game around with extrodinary team fights in the end. Therefore, our model could only use early game data for a general insights. There are miracles in LOL Esport match and that is why audience love to watch the games.

There are ineffective features that could not shift the game result significantly. For example, first dragon is not as important as the others. However, I do believe it offers an insight for more uses such as strategies and team composition.

Conclusion: In the study, I obtain the dataset from Oracle Elixir and clean the data for the most important factors in the early games. By building Random Forest Classifier pipeline, I reach the accuracy of 0.747. This means that game is already shift significantly to the team with better early advantages. This study could provide a general prediction for team with the same level skill. However, there are other factors could affect the game outcome due to the dynamics of the game. The first baron is spawn on 20 minutes into the game, which could shift the win rate significantly. In addition, champions and team compositions are built differently: some focuses on early advantages and some focus on late-game team fights. Therefore, if early advantages team could not obtain enough advantages, the win rate could shift drastically.