**Big Data Analytics Final Report Template**

Tong Liu, Kaiji Lu, Rena Ren

tl2871, kl3065, yr2325

Columbia University

e-mail: [tl2871@columbia.edu](mailto:tl2871@columbia.edu), [kl3065@columbia.edu](mailto:kl3065@columbia.edu), yr2325@columbia.edu

*Abstract*—**The report tackles the big problem of lack of in-game odds calculation which is prominent in the E-sports betting industry. We used data provide by Riot Company from 50000 ranked game of League of Legends, and used three different classification models--GBT classifier, decision tree classifier and logistic classifier—to evaluate how different objectives in game will affect game result. These classifiers appear to work equally good for the data with error rate at around 3.5% percent. Among the result, we also find that when inhibitor take downs and tower take downs happen, they have more influence on the result of the match.**

# Introduction

E-sports has caught capital’s attention the past 5 years due to the large volume of its viewership and its potential to generate profit. Among all, E-sports betting has been an inevitable hot topic of discussion. Researches estimated that E-sports wagering is expected to reach 13 billion in year 2020. Some large betting websites such as pinnacle has already seen 300% year over year E-sports betting volume growth. In addition, E-sports has also entered the main stream, proving its importance gradually. Among all the E-sports titles, League of Legends betting has received most attention because of its worldwide popularity.

However, there’s one blank spot that most traditional sports betting has but not E-sports- that is in-play betting. Different from pre-match betting, In-play betting means players are able to bet during the match. This new type of betting has revolutionized traditional sports betting in the past, growing the market several times over in the space of a few years.

In this project, we are going to solve the problem of in-game prediction of League of Legends matches, providing a win-lose prediction of two teams during a real time match. This will pave the way for odds calculation of in-game E-sports betting.

# Related Works

Multiple big betting websites such as Pinnacle.com, William Hill and Betway have in-house odds calculating machine and follows this discipling [1]. However, there E-sports betting sector is limited by in-game betting. So far, there are also no third-party odds calculating company that has in-game E-sports odds services.

# System Overview

Our approach for this problem is as follows：

1. Data processing: gather enough match make data and find relevant features that influences the result of the match
2. Training: Use various classification models to train and verify a model that’s most suitable for our problem.
3. Testing: Feed real time match data into the dash board and predict winning team.

Data processing: We selected 50000 ranked games data of League of Legends. This dataset was collected using the Riot Games API.

The data contains fields of

1. Game ID
2. Creation Time (in Epoch format): when this data is created
3. Game Duration (in seconds): the length of the game.
4. Season ID: which season this game happened during.
5. Winner (1 = team1, 2 = team2): the result of the game
6. First Baron, dragon, tower, blood, inhibitor and Rift Herald (1 = team1, 2 = team2, 0 = none): these are objectives in the game. When one team captures one or more of these objectives, they will be rewarded a significant amount gold which in turn will increase their probability to win.
7. Champions and summoner spells for each team (Stored as Riot's champion and summoner spell IDs): choices that players make before the game starts. Note that some Champions have significant higher/lower win rate according to OP.GG [2].
8. The number of towers, inhibitor, Baron, dragon and Rift Herald kills each team has: objectives that will grant team gold in the game.
9. The 5 bans of each team (Again, champion IDs are used): what champions the other team are banned from using before the game starts.

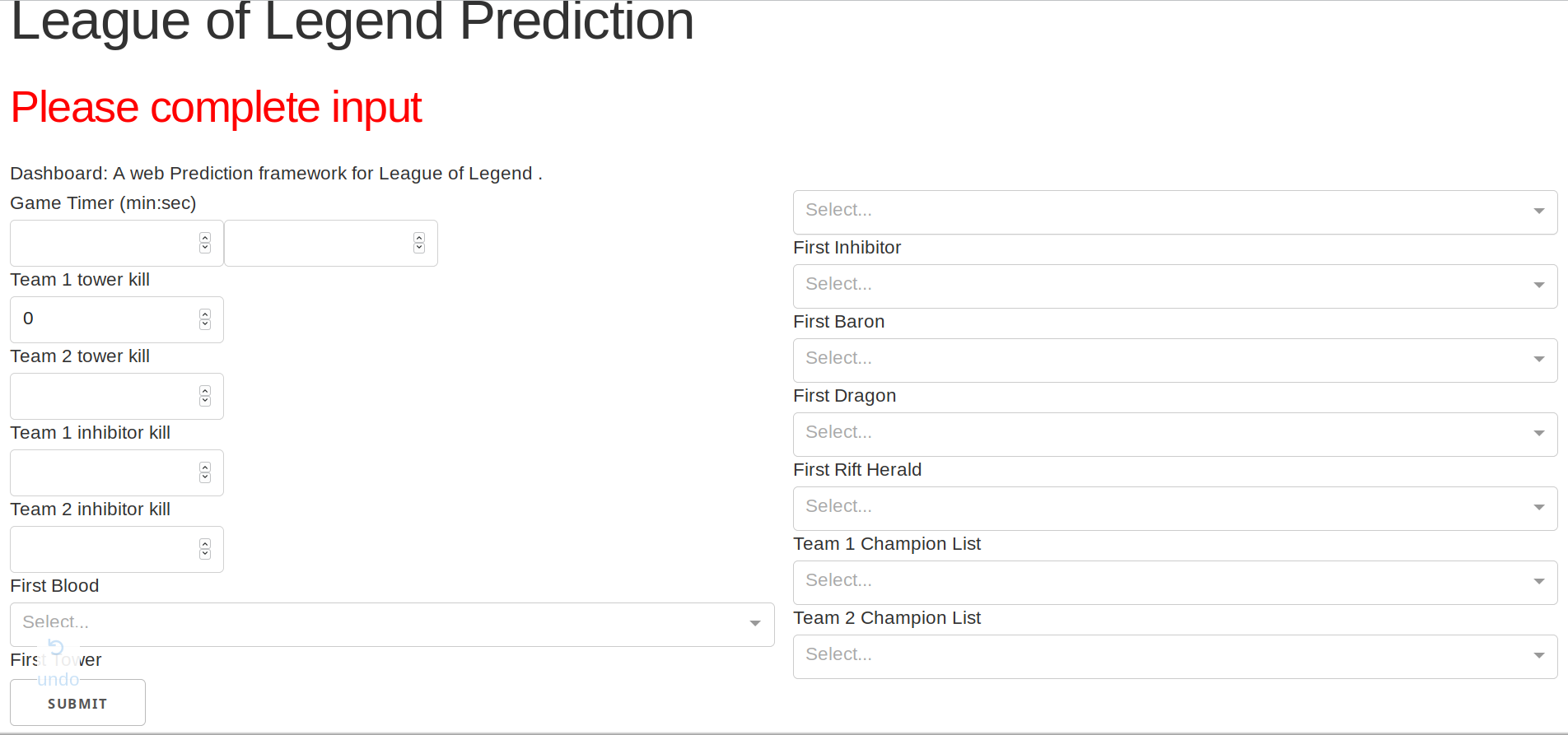
Note that these are not independent data. Each field might influence the game differently when combined with each other. We abandoned Creation time, Season ID and bans of each team because they are not relevant.

Next, we make Winner the label and all other data the features. Later we transformed csv data into libsvm format to make use of Apache Spark.

After that, we transformed the data into vectors, and split the data into training data and testing data. The ratio of the split is 7:3 with 7 being the training data and 3 being the testing data.

Training: we used three classification algorithms to train our data.

Testing: We developed a dashboard which enables users to input real time data and output winning team.



# Algorithm

We use Spark to train our data. Apache Spark is an open-source distributed general-purpose cluster-computing framework. It enabled us to analyze our data without looking into the algorithm details.

1. GBT classifier

Gradient boosting produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

1. Decision tree classifier

Decision tree learning uses a decision tree (as a predictive model) to go from observations about an item (represented in the branches) to conclusions about the item's target value (represented in the leaves). It is one of the predictive modelling approaches used in statistics, data miningand machine learning. Tree models where the target variable can take a discrete set of values are called classification trees; in these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels.

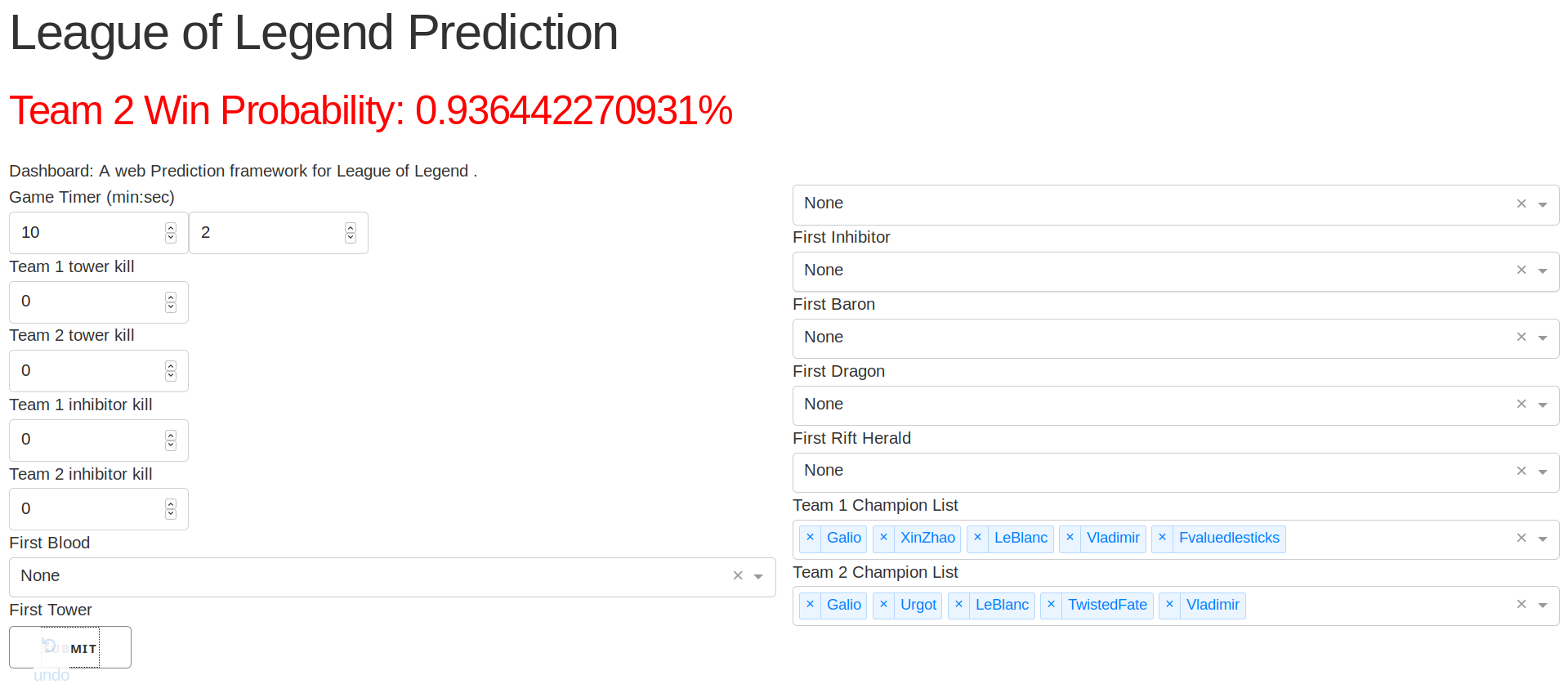
1. Logistic regression

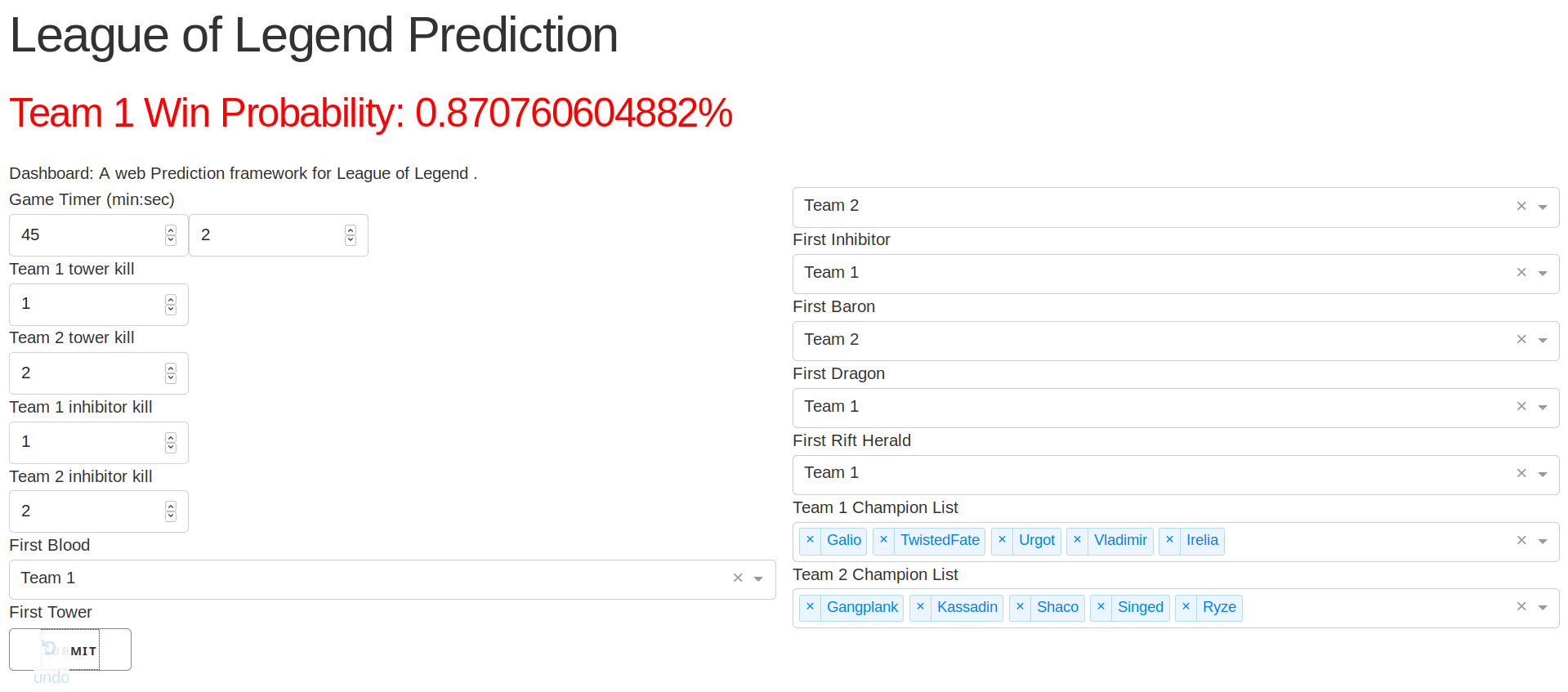
the logistic model is a widely used statistical model that, in its basic form, uses a logistic function to model a binary dependent variable; many more complex extensions exist. In regression analysis, logistic regression (or logit regression) is estimating the parameters of a logistic model; it is a form of binomial regression.

# Software Package Description

We used Dash by Plotly to develop our frontend prediction machine. It is a productive Python framework for building web applications. Dash abstracts away all of the technologies and protocols that are required to build an interactive web-based application, which is vastly convenient for a project like ours.

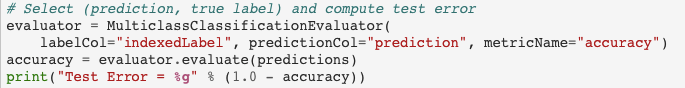
Below are a few UI and user cases of our prediction machine. We input some features like champion selections, towers and inhibitors and get a win rate for a prospective winning team.





# Experiment Results

The error for our GBT classifier is 3.33%. We used Spark’s evaluator to evaluate the accuracy.



The error for our decision tree classifier is 3.87%. We used similar approach to evaluate the accuracy.

The error for our Logistic regression is 3.78%. Approaches are similar to the previous two algorithms.

These means that the three algorithms do not differ much in terms of performance. Further investigations also reveal that different features contribute differently to the winning results. Among all the features described in the dataset section, the result is more sensitive to the change of First Tower and First inhibitor, and less sensitive to other features. Investigations also show that duration of game seem to have little effect on the result of the game. Possible reason might be that the data riot provided to us is end game data. If we are able to obtain when each critical objectives happened during the game, aka the time each data is collected, we might be able to find how time, combined with other features, can influence the game result.

# Conclusion

After building model on League of Legends ranked game data, we realized that different classification model worked similarly on the dataset. Among all the features that can influence a game’s result, Inhibitor and Tower are the most sensitive features. In future works, we would like to further investigate if there exists linear or non-linear relationships between the probability of winning and features( ie. Win rate and time first blood happens, win rate and gold amount). With those in mind, we can build a more complex in game odds calculating machine that will finally be commercially useful.

##### Acknowledgment and contribution

we would like to thank Professor Lin for the great semester and all the TAs for the hard work. Below is a contribution distribution to the project.

Tong Liu: 1/3, dashboard and front end

Kaiji Lu: 1/3, data processing and training

Rena Ren: 1/3, data selection and report

##### Appendix

##### References

[1] https://www.investopedia.com/articles/dictionary/042215/understand-math-behind-betting-odds-gambling.asp

[2] http://na.op.gg/champion/statistics