ORIE 4741 Midterm Report:
Can We Predict LOL Game Results?

Yuhan Li, Anqi Ren, Jiang Zhu

1 Introduction and summary

The ultimate purpose of this project is to identify the most relevant features that lead to a win in a LOL game match using a dataset from Kaggle. The project could help common players to improve their gaming skills and help professional teams to develop their strategies. This midterm report mainly focuses on data cleaning, feature selection, model analysis, and plans for future work. Regarding to feature selection, 8 out of 11 features are selected based on R^2 of the linear regression model. Three models, linear regression, classification tree, and SVM, are discussed in this report. Based on analysis, the classification tree is the optimal model among three options.

2 Data

2.1 Data Cleaning

In order to further explore the relevant features of predicting the response, the team first cleaned the data to narrow the range of selection. The original data set stores data per row using gameID as the primary key, which means each row has game data for both winning team and losing team. In this case, gameID, game duration, and seasonID are irrelevant. To speed up analysis, the team selects corresponding variables for winning and losing team separately, so each row in the data set after cleaning only has game data for winning team or losing team.

2.2 Data Description

Features	Data Type	Explanation
win	Binary	Win (1) or lose (0)
firstBlood/firstTower/firstInhibitor/firstBaron/firstDragon/firstRiftHerald	Binary	Whether the team got first blood, first tower, first inhibitor, first baron, first dragon, first rift herald (1) or not(0)
towerKills/inhibitorKills/baronKills/dragonKills/riftHeraldKills	Integer	Number of towers/ inhibitor/baron/dragonKills/riftHeraldKills killed in the game
t_champ1id	Integer	Which champ team members used
t_ban	Integer	ChampIDs banned by the team

Figure 1: The data column win is the dependent variable and the rest of variables are possible features used to predict the response. The cleaned dataset consists of 102980 rows and 31 columns.

2.3 Descriptive Statistics

Figure 2 below show the correlation of win/loss with 11 features. The first plot indicates the effects of first Blood/Tower/Inhibitor/Baron/Dragon/RiftHerald on wins. First inhibitor has the most effect on wins. The other plots shows the relations of wins to the frequency of killed Tower/Inhibitor/Baron/Dragon/RiftHerald. The point size represents the frequency of each specific kill numbers. Winning teams tend to destroy as much towers as possible. For inhibitor, baron and dragon, winning teams kill more of these elements than losing team, but some teams still lost when they kill many of inhibitors, barons and dragons. Killing the riftHerald does not guarantee victories either.

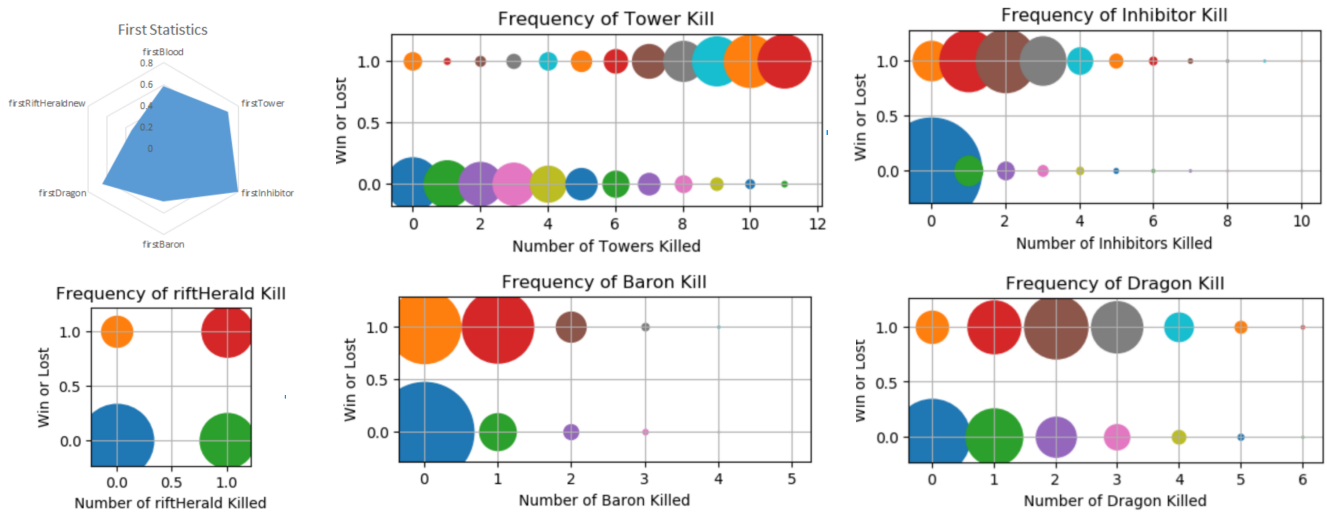


Figure 2: Descriptive Statistics

3 Data Analysis

3.1 Feature Selection

First, we ran simple linear regressions on all 11 features and compare R^2 of these models and choose the features with $R^2 > 0.1$ (the higher the R^2, the more relevant the features are to the dependent variable). The project will also test the effectiveness of how many features to select in Section 3.3. Figure 3 shows the result of SLR.

Features	tower Kills	firstIn hibitor	inhibit orKills	dragon Kills	firstTo wer	firstBa ron	baron Kills	firstDr agon	firstRif tHeral d	riftHer aldKill s	firstBl ood
R2	0.6064	0.5289	0.4290	0.2350	0.1654	0.1639	0.1474	0.1203	0.0494	0.0494	0.0325

Figure 3: The R^2 of win vs 11 features. The project will use the 8 features in the following models.

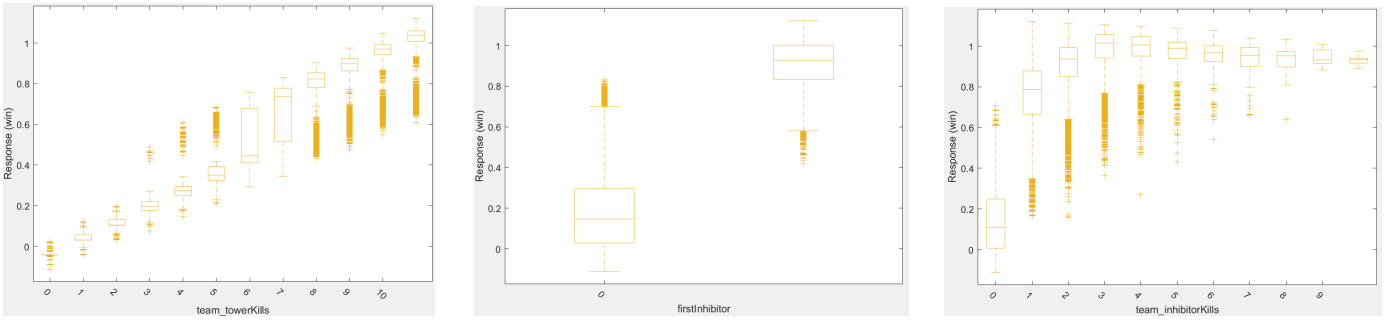
3.2 Model Selection

As a preliminary test, the project tested the following three models. Figure 4 show box plots of partial results from linear regression. As we can see from Figure 4, there are clear trends that winning probability will increase as towerKills/inhibirtorKills increase and with getting the first inhibitor. Figure 5 show the Confusion Matrix of true y vs predicted y for Model 2 and 3. For both models, the misclassification rates are pretty low, especially in predicting a winning game.

–*Model 1: Linear Regression.* Regress win linearly on 8 features: towerKills, firstInhibitor, inhibitorKills, dragonKills, firstTower, firstBaron, baronKills, firstDragon.

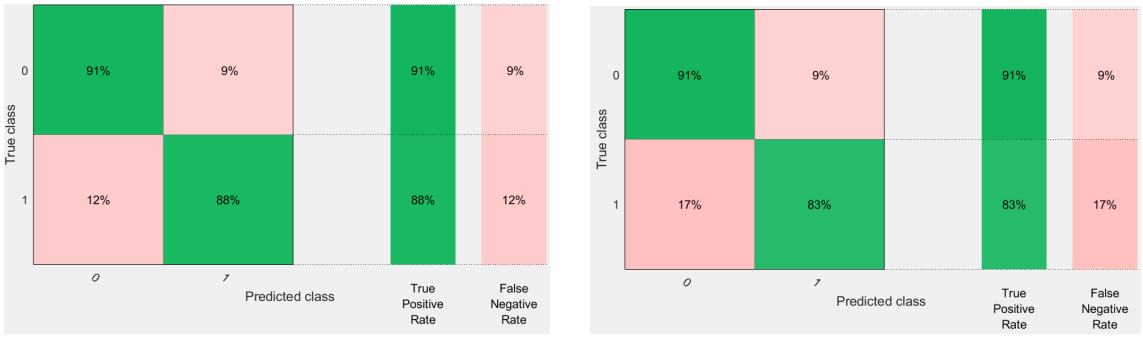
–*Model 2: Classification Tree.*

–*Model 3: Linear Support Vector Machine (Linear SVM).*



(a) towerKills vs win (b) firstInhibitor vs win (c) inhibitorKills vs win

Figure 4: Model 1: Linear Regression



(a) Model 2: Classification Tree (b) Model 3: SVM

Figure 5: Confusion Matrix for Model 2 and 3

3.3 Preliminary Analysis

- How to test the effectiveness of the models?
  - *Cross-validation:* The project used cross-validation to check the effectiveness of feature selection. We randomly divided data into training and test sets. For feature selection, we left out zero, one, two or three features in turns and compared the the overall error. It turned out that the combination of 8 features gave the smallest misclassification error.
  - *Mean Square Error:* The project computed the mean square error to check the effectiveness of each model. Both through numerical computations (accuracy= 89.8%) and through confusion matrix in Figure 5, we can see that Model 3 is the best model so far.
- How to avoid underfitting and overfitting?
  - To prevent underfitting, the project are using large amount of data. Additionally, we are planning to add more champion related features such as champion selection and ban .
  - To prevent overfitting, the project used a k-fold cross validation ($k = 5$ in our preliminary models). We are also planning to use Lasso regression and l1 regularization to test the relevance of each feature on the dependent variable and perform feature elimination if needed.

4 Future Plans

Since the dependent variable is a binary variable, classification models will more likely predict the results better. The project will in the future consider using logistic regression models, nearest neighbor classifiers and more variations of SVM. In addition to test more models, the project will add champion related features into the models including champion selection and ban.