Classification of League of Legends (LoL) Results

Pan Shuang

1. Problem Introduction

LoL is one of the most popular online game currently around. It is a 5 vs. 5 competitive game. The target of the game is to destroy the base of the enemy. The team which grabs more resources, such as Dragon and Baron, is more likely to win the game.

Predicting its result from the game status is the target of this project. Now a dataset containing approximately 5 million game records is given. About 3 million is used for training and 2 million for testing. The record of a specific game includes the creation time of the game, the game duration, the ID of the season, the winner team, the team which gets first blood, the first tower, the first inhibitor and so on.

Some classifiers, which use information other than the winner to predict the winner, are expected to obtain. The algorithms used in this project include Decision Tree (DT), Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Random Forest (RF) and Voting Ensemble. In addition, Grid Search algorithm is used to get the optimized parameters for some classifiers.

2. Algorithms

2.1 Feature Selection

Consider the practical implication of the data, the gameId, creationTime and winner are not selected as the features to be used. In addition, it is noticed that the feature seasonId of all the data are the same (=9), so it is not selected too. The correlation coefficients (CC) of each feature and the winner are calculated and the results are shown in Table 1. Only the features with its CC's absolute value greater than 0.15 are chosen.

Table 1 CC and feature selection table

$Feature\ name$	CC	$Chosen?$	$Feature\ name$	CC	$Chosen?$
$gameId$	0.0172	×	$t1\_towerKills$	-0.7745	√
$creationTime$	0.0174	×	$t1\_inhibitorKills$	-0.6519	√
$gameDuration$	0.0211	×	$t1\_baronKills$	-0.3750	√
$seasonId$	0	×	$t1\_dragonKills$	-0.4722	√
$winner$	1	×	$t1\_riftHeraldKills$	-0.2218	√
$firstBlood$	0.1740	√	$t2\_towerKills$	0.7840	√
$firstTower$	0.3747	√	$t2\_inhibitorKills$	0.6578	√
$firstInhibitor$	0.5342	√	$t2\_baronKills$	0.4002	√
$firstBaron$	0.2609	√	$t2\_dragonKills$	0.4925	√
$firstDragon$	0.3083	√	$t2\_riftHeraldKills$	0.2281	√
$firstRiftHerald$	0.1193	×			

The reduction on data's dimension is of great significance because it can speed up the training process of some classifiers.

2.2 Support Vector Machine

SVM is a powerful machine learning algorithm which can be used for classification. It finishes the classification by first generating hyperplanes iteratively that segregates the classes. Then, it will choose the hyperplane that separates the classed correctly. The key of the SVM is its kernel function, which can transform an input data space into the required form. The SVM is considered to be used because it is powerful at handling the binary classification.

2.3 Multi-Layer Perceptron

MLP is a feedforward artificial neural network (ANN) that generates a set of outputs from a set of inputs. It has several layers connected as a directed graph between the inner layers. MLP uses backpropagation for training the network. The MLP is chosen to be used because it is suitable for nonlinear classification.

2.4 Decision Tree

DT is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label. DT is used because it performs well when the features of the samples are explainable and DT is relatively easy to optimized compared to other classifiers.

2.5 Random Forest

RF is an ensemble of DTs, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result so it can be viewed as the advanced version of DT.

2.6Voting Ensemble

Voting is a common ensemble method. It has the hard type and soft type. The hard voting ensemble involves summing the votes for crisp class labels from other models and predicting the class with the most votes. The soft voting ensemble involves summing the predicted probabilities for class labels and predicting the class label with the largest sum probability.

2.7 Grid Search

Grid Search is essentially an optimization algorithm which lets you select the best parameters. The possible values of each parameter are arranged and combined, all possible combination results are listed to generate "grid", and then each combination is used for classifiers training, and the performance is evaluated by cross validation.

The following table shows the optimized parameters obtained from Grid Search algorithm and they are used for the application of the SVM, MLP, DT, RF and ET in this project.

Table 2 The optimized parameters obtained by Grid Search algorithm

$Algorithm$	$Parameters$
SVM	$gamma = 0.1$
MLP	$hidden\_layer\_sizes = (10, 10)$
DT	$criterion = 'entropy'$ $max\_depth = 7$ $min\_samples\_split = 15$
RF	$n\_estimators = 13$ $max\_depth = 10$ $min\_samples\_split = 2$ $bootstrap = True$
$Voting$	$voting = 'soft'$

The algorithms part can be summarized as Figure 1.

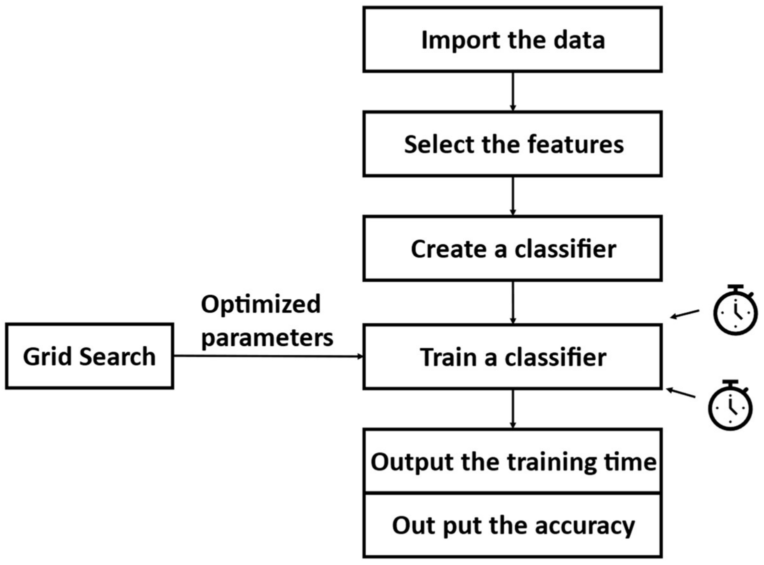


Figure 1 The algorithm flowchart of a classifier

3. Experiment Requirements

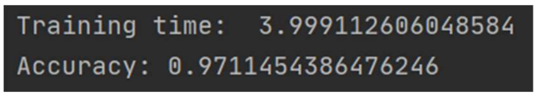
The conversion of the integrated development environment (IDE), interpreter and the main packages used in the experiment are shown in the following table. The experiment can be conducted in $Jupyter\ Notebook$ ($Anaconda\ Navigator$ 1.9.12) too.

Table 3 The versions of the IDE and packages of the experiment

	$Pycharm$	$Python$	$sklearn$	$pandas$	$numpy$
$version$	2020.1	3.8.2	0.23.1	1.0.5	1.19.0

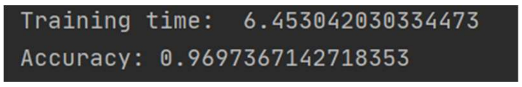
4. Experiment Results

The screenshots of the experiments' results are shown below.



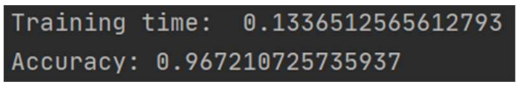
```
Training time:  3.999112606048584
Accuracy: 0.9711454386476246
```

Figure 2 The result of SVM algorithm



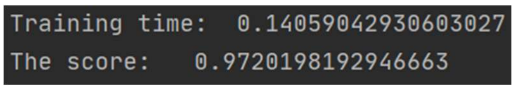
```
Training time:  6.453042030334473
Accuracy: 0.9697367142718353
```

Figure 3 The result of MLP algorithm



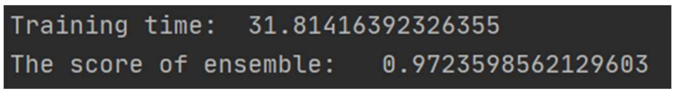
```
Training time:  0.1336512565612793
Accuracy: 0.967210725735937
```

Figure 4 The result of DT algorithm



```
Training time:  0.14059042930603027
The score:   0.9720198192946663
```

Figure 5 The result of RF algorithm



```
Training time:  31.81416392326355
The score of ensemble:   0.9723598562129603
```

Figure 6 The result of Voting algorithm

These results are summarized in the following table.

Table 4 The training time and accuracy of each algorithm

$Algorithm$	$Training\ Time\ (s)$	$Accuracy$
SVM	3.9991	0.9711
MLP	6.4530	0.9697
DT	0.1337	0.9672
RF	0.1406	0.9720
$Voting$	31.8142	0.9724

5. Comparison and Discussion

5.1 Simple comparison

From the Table 5, some conclusions can be drawn. The classifier which needs the least training time is DT but its accuracy is the lowest. The classifier which gets the highest accuracy is Voting but it need the longest time to train. It is obvious that the training time of DT and RF are both much lower than other classifiers. Consider both training time and accuracy, the RF is the best classifier to solve the classification of LoL results problem.

5.2 Further Discussion

The reasons why the training time of the classifier other than DT and RF are much longer are as follows. When the dimensions and amounts of data is very large, SVM and MLP need much time to train. For SVM, it needs to calculate every data point at the same time in order to find the suitable hyperplane. In addition, the kernel function for SVM may not be suitable so the hyperplane is difficult to find in the new space. For MLP, it puts the data into the network one by one to train. Additionally, the number of MLP’s nodes are large, some nodes’ weights are difficult to train, because they almost are not affected by the output layer during the backpropagation process. For the Voting classifier, because it needs to train all its component classifiers, it needs the longest time to train.

The reasons why RF performs well are as followers. Firstly, the features of the data in this problem have obvious meaning and tree classifiers are good at handling reasonable problems. A lot of gamers say, “League of Legends is a tower push game.” The ultimate goal of one team is to destroy all the towers of the other team, so the tower kills data directly reflect the situation of the game. Figure 6 shows the data point with the tower kills of team 2 and the final winner. The correlation coefficient of the two feature is 0.784, which means they are highly related. A tree with only three layers is train and shown in Figure 7. The nodes using tower kills data take the big part and it is very reasonable.

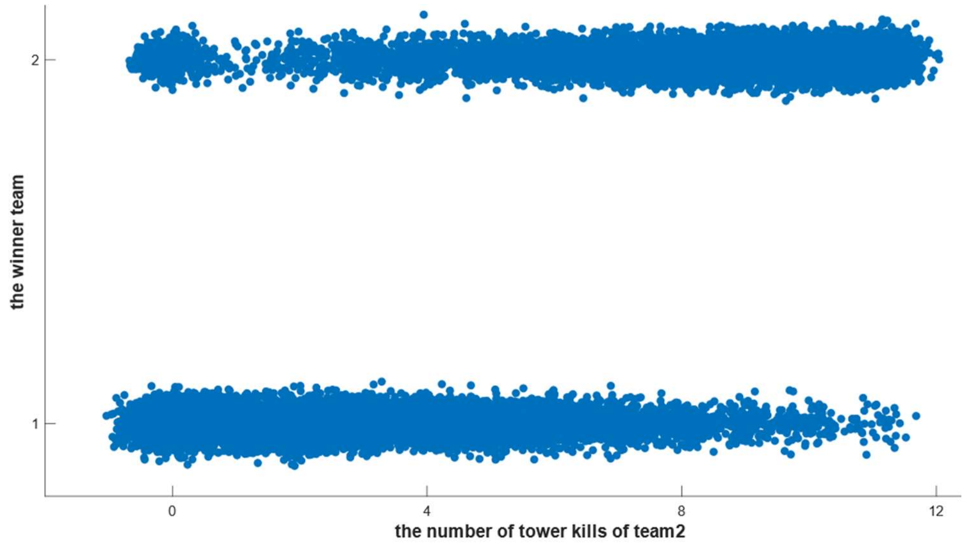


Figure 7 The relation between the tower kills of team 2 and the final winner

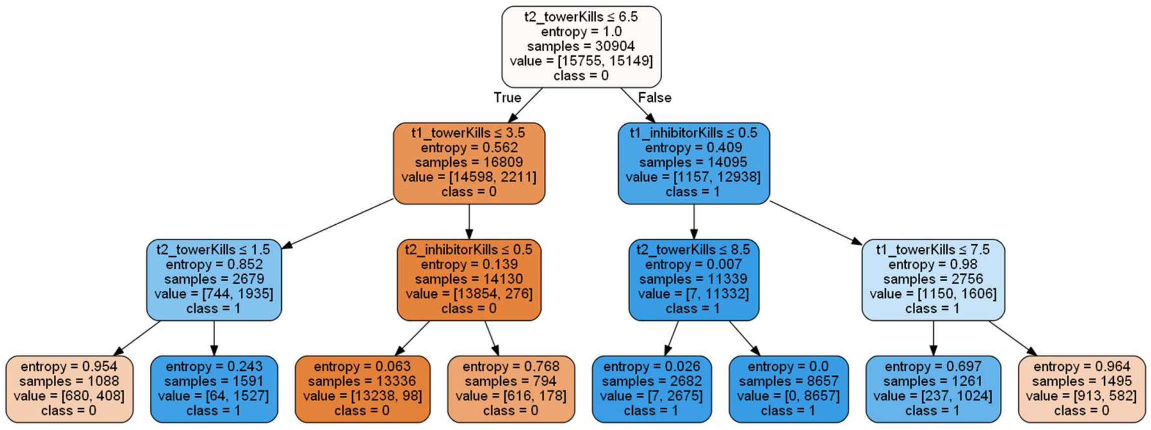


Figure 8 A DT with three layers

Secondly, the maximum depth of the RF is limited to 10, which controls its complexity. Figure 8 shows the relation of the maximum depth of RF and its accuracy. It shows that as the maximum depth of the tree increase, the accuracy rises first and then falls. It rises first because the shallow trees can't distinguish the samples subtly and it then falls because the forest is too complex so that it overfits the training set. The limitation of RF's depth can prevent it from overfitting and make it have good generalization performance.

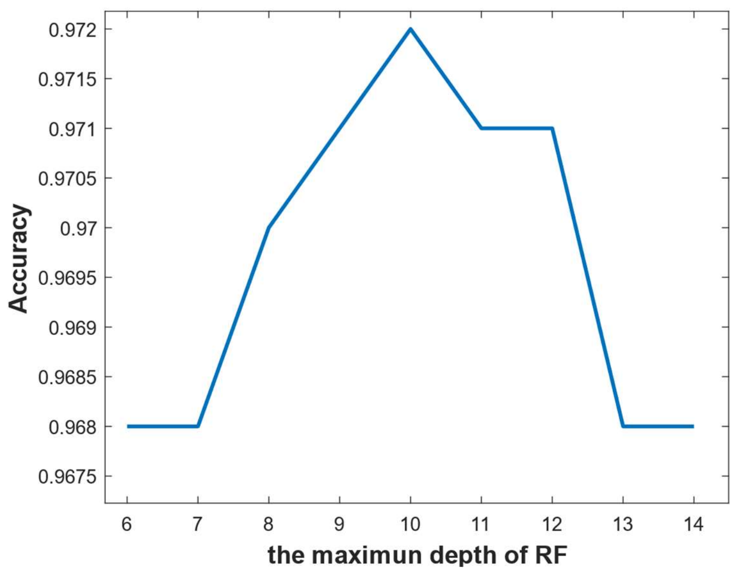


Figure 9 The relation curve of max_depth and accuracy

The tree classifiers also have their drawbacks. They don’t pay much attention on the connections among the features. They always focus on a specific feature at a node. In fact, the connection of features in this problem is strong. For example, in this problem, the team gets the first Rift Herald is easier to kill more towers, because the Rift Herald can make a strong damage on enemy’s towers. However, the RF combines a certain number of trees, so it overcomes this shortcoming to some extent.

5.3 Future Work

The future improvement can be made in finding a good kernel function for SVM or directly write a kernel matrix. There is still a great number of parameters in MLP that can be optimized. In addition to RF, Extra Tree is also an ensemble method about tree that can be tried. What’s more, there are some other big data method to try, for example, K-Nearest Neighbor, which is very suitable for classification problem.

6. Summary

The goal of this project is to get some classifiers that can classify the LoL results well. The feature selection is done first. Then, SVM, MLP, DT, RF and Voting Ensemble classifiers are trained to solve the problem. Grid Search method is also used to find the optimized parameters for classifiers. The experiment results are analyzed and the

RT classifier is thought to be the best suitable one for handling this problem and the advantages of is are analyzed too. All classifiers have their drawbacks and they can be improved in future. The experience I got from this project is that we should pay attention to the unique characteristics of classifiers so that we can select the best targeted classifiers in practical applications.

References

1. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
2. https://www.kaggle.com/ahmethamzaemra/mlpclassifier-example