Project 1

REPORT

(Prediction model for the results of League of Legends(LOL) by ANN classifier and DT classifier.)

1. Introduction :

League of Legends game is the most popular online game of team fighting, each game has two teams for real-time fighting, through tower dismantling, playing wild monsters and killing the enemy side hero to gain an advantage, the first to dismantle the enemy headquarters as the victory. In LOL(League of Legends), we can often predict the winner of a game by analyzing the battle status of the teams. The purpose of this report is to use the classifier to classify the situation and predict the outcome of the match.

This time, I used two methods, ANN and decision tree, to train the training set data. By analyzing and classifying more than a dozen factors such as the length of the competition, the first drop of blood, the first tower and so on, I got an algorithm model, and tested the generated model with "test data" to obtain the accuracy of the model.

2. Algorithms :

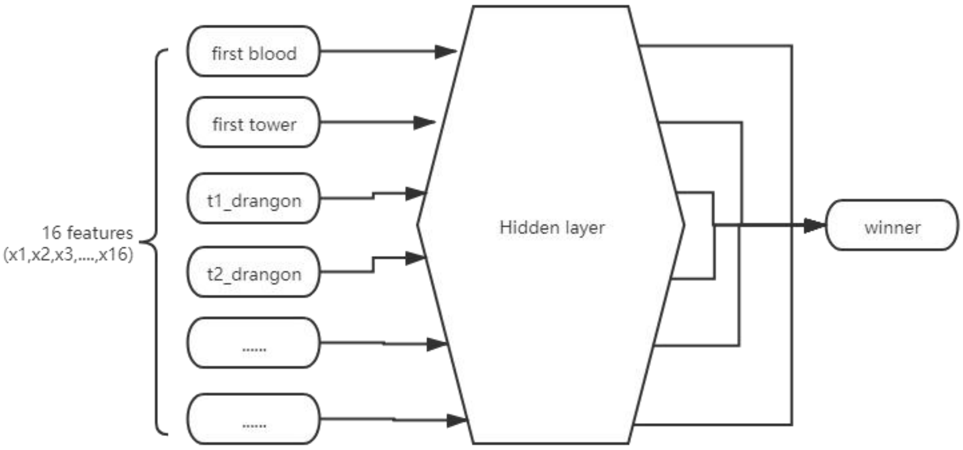
In this project, I used the match records of approximately 3 million single-player players as training sets. Each record is all publicly available game statistics for a game played by some players, including an important area called "winner." The winner of the game is assigned to a class named "1" and the loser is named "0".

The first method: Artificial Neural Network (ANN):

Artificial neural network simulation is divided into two stages: Firstly, training and learning: According to the provided data and a certain training algorithm, the relationship between input and output is found, and the connection weight of each processing unit is adjusted to achieve a certain training accuracy, and the network training is completed. Secondly, after simulation and information input, the trained network can provide satisfactory results according to the learning and training results.

First, the program reads the data contained in the two .csv files to generate training set data and test set data. Then, use machine learning to classify the training set data and generate a neural network model.

The model generated is structured by Multilayer Perception(MLP), which is a forward structure of artificial neural network, mapping a set of input vectors to a group of output vectors. There are three layers--Input layer, hidden layer, output layer.(table 1)



(table1)

After model is set up, it should be tested by the test set data to get the accuracy of it. The 16 features of the test set data are the input data and the model output the results which are compared with the results in test set data.

Model training has criterion creation which is a loss measurement and optimizer creation which optimizes for the model. This comparison is much more easy which just distinguish class “0” and class “1”. Finally, the process output the accuracy of the model which is set up by training set data and tested by test set data.

The requirements needed in this method are torch, pandas, numpy and sklearn.

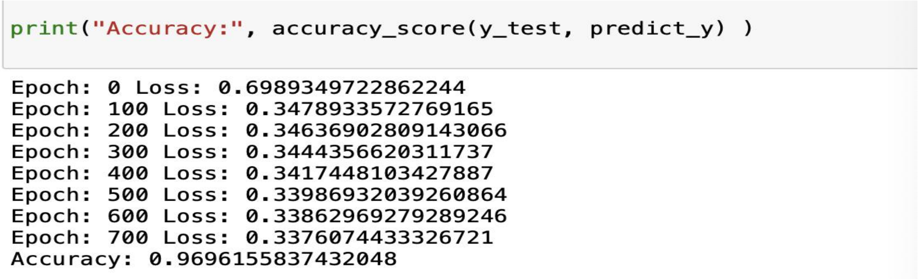
Here the process run several times to get the best value of epoch. (table 2)

epoch	accuracy	time
0	0.9647	2.45’
100	0.9652	2.27’
200	0.9668	2.68’
300	0.9706	2.88’
400	0.9735	2.96’
500	0.9726	3.12’
600	0.9707	3.20’
700	0.9698	3.38’

(table 2)

Epoch=400 was chosen, so the training accuracy on average is 0.96995.

Here is the results processed in Jupyter(table 3):



```
print("Accuracy:", accuracy_score(y_test, predict_y) )
```

```
Epoch: 0 Loss: 0.6989349722862244
Epoch: 100 Loss: 0.3478933572769165
Epoch: 200 Loss: 0.34636902809143066
Epoch: 300 Loss: 0.3444356620311737
Epoch: 400 Loss: 0.3417448103427887
Epoch: 500 Loss: 0.33986932039260864
Epoch: 600 Loss: 0.33862969279289246
Epoch: 700 Loss: 0.3376074433326721
Accuracy: 0.9696155837432048
```

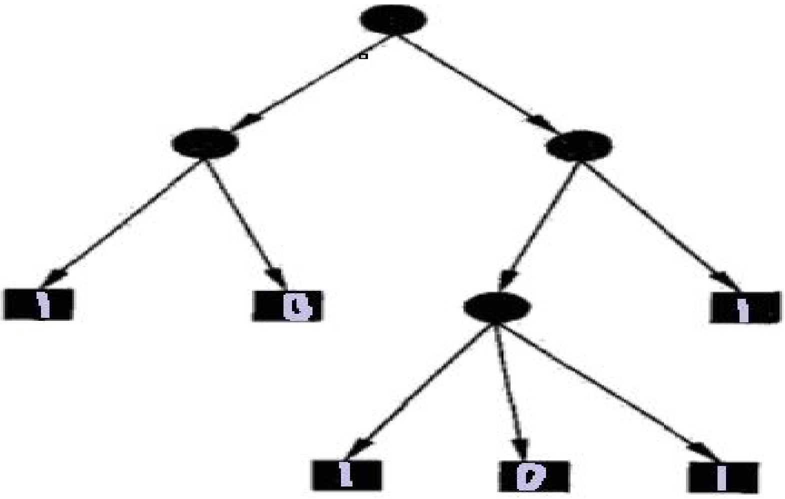
(table3)

(The accuracy and the rate is not fixed, they fluctuate slightly.)

The second method: Decision Tree (DT): Decision tree is a nonparametric supervised learning method, which is mainly used for classification and regression. It can extract a tree type classification model from a given unordered training sample. It is composed of nodes and directed edges. Generally, a decision tree contains a root node, several internal nodes and several leaf nodes. The decision process needs to start from the root node. The data to be tested is compared with the characteristic modes in the decision tree, and the next comparison branch is selected according to the comparison results until the leaf node is taken as the final decision result.

In this project, the decision tree generation functions already available in Python. First, the program reads the data contained in the two csv files to generate training set data and test set data. Then, use machine learning to set a decision with the training set data.

Here is a part of the decision tree in this model. (table4) In it, “0” is the class if the winner id team 1 and “1” is the class if the winner is team 2.



(table 4)

The requirements needed in this method are pandas, numpy and sklearn.

Here are the 10 times test for the accuracy of the decision tree.(table 5)

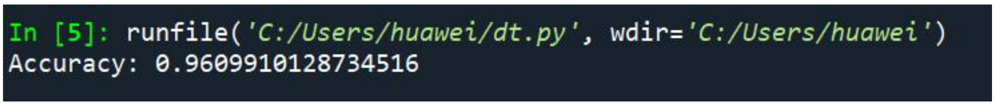
test	accuracy	time
1	0.9721	1.59’
2	0.9568	1.65’
3	0.9596	1.71’
4	0.9735	1.96’
5	0.9667	1.85’
6	0.9595	1.63’
7	0.9732	1.79’
8	0.9798	1.88’

9	0.9683	1.62′
10	0.9624	1.51′

(table 5)

So the best accuracy is 0.9798 and the average accuracy is 0.96719.

Here is the result of decision tree model obtained by the program using Spyder (table 6):



```
In [5]: runfile('C:/Users/huawei/dt.py', wdir='C:/Users/huawei')
Accuracy: 0.9609910128734516
```

(table 6)

3. Comparison :

The accuracy of the method of ANN is 0.9669 and the accuracy of the method of DT is 0.96719. By comparing the accuracy, it can be seen that the accuracy of the two methods is almost the same and very high, and the prediction models generated by the two classifiers are excellent. However, from the perspective of model running time, the model generated by ANN classifier consumes about 3 seconds in the prediction process, while the DT method consumes about 1.3 seconds. The former consumes much more time than the latter, so the DT model is more excellent in this classification model.

4. Conclusion :

Both models are highly accurate and can be applied to predict the final outcome of league of Legends games. However, both of these two methods have the same problem. Since the training set data may not contain all the situations, the models cannot avoid the occurrence of unexpected situations. Sometimes an eigenvalue will be different from the predicted situation, which will lead to a great increase in the error rate of the predicted situation. In addition, machine learning is a prediction model based on the analysis and classification of objective data. It may be interfered by some factors, and the prediction is biased.

But in general, these two prediction models can better predict league of Legends games without the above two less cases.