Classification of unlabeled LoL match records with “Win/Loss”

从凯宣

October 10, 2020

1.Introduction:

1.1 The brief introduction of classifier algorithm in supervised learning

As an important member of supervised learning, the classification algorithm can be trained based on the existing labeled samples to form a accurate classification model. The trained classification model can make an accurate category judgment for the unknown class instance based on its existing features.

1.2 The objectives and requirements of the project

1.2.1 Training set:

In this project, the training set that we will use has access to about 30,000 match records of solo gamers of League of Legends. Each record is looked as a training sample that having 21 attribute fields including an label field called "winner". If "winner" is “1”, the team 1 won the match, or vice versa. Additionally, some of the attribute fields may be useless.

1.2.2 Objectives of the project:

Use the useful features in the training set to train the classifier to form a correct classifier model. Then make a relatively correct prediction on the test set.

1.2.3 Requirements of the project

What requirements we must satisfy is at least one classifier is used, and the prediction accuracy of the test set is at least 50%. In addition, evaluate the advantages and disadvantages of one or more classifiers you choose.

1.3 Brief description of methodology

1.3.1 Data preprocessing

1.3.1.1 Duplicate Data

Training set may include data objects that are duplicate of another. If you train the classifier using the data duplicate of another, you may put in the time on the training and get little or nothing. Thus, detecting the number of data duplicates and deleting them is necessary.

1.3.1.2 Data standardization

For the purpose of avoiding the effect that the scope difference between each attribute produce, It is important that do the data standardization on training set and testing data to make the characteristic value of each attribute in the same range.

1.3.1.3 Correlation between attribute and label

Not every attribute values has a correlation with label. Sometimes, a sample will have some attribute values that have no obvious relationship with the label. If the classifier also uses these attributes as input attributes, the final model we trained may be inaccurate.

Therefore, I will analyze the correlation between the attribute value of each column and the label value below by calculating the correlation coefficient between the two columns, and I will delete some columns corresponding to the attribute values with very small correlation coefficients.

1.3.2 Classification algorithm

After data preprocessing, I got 30,000 sample data with 16 features and one label.

1.3.2.1 Decision tree：

We first use the decision tree algorithm as the classification algorithm, which can use the characteristics of the training set to form a tree diagram. Then, when predicting the test set, we will make an attribute judgment on the sample at each node of the tree diagram until we make a judgment on the class of the sample.

1.3.2.2 Artificial neutral network：

I chose the multiply neural network as the basic model of the classifier to perform classifier training and sample prediction. The neural network I built here has two layers, first the first convolution layer. This layer uses linear mapping and uses the sigmoid activation function to map 16 features into 6 features; and the output layer layer linearly maps 6 features into two features and output. Finally, we compare the magnitude of the two output values to get the category.

1.3.2.3 K-Nearest Neighbor

Finally, I chose the K-Nearest Neighbor algorithm. In this algorithm, we are equivalent to establishing a 16-dimension coordinate system. At this time, the training process is equivalent to converting each training sample into a specific point in the 16D feature space

according to the attribute value of each training sample. When performing sample prediction, we place the test sample in this 16-dimensional coordinate system, and look for the K nearest labeled training samples around it, and then this test sample will be labeled as the class that corresponds to K nearest samples the most in the K nearest training samples.

2.Algorithms:

2.1 Data preprocessing:

Name	type	Meaning or performance
df	variable	Store all of the training data including features and label
df.duplicated()	function	Check whether each line is repeated in df, if there are duplicate lines in the line, return "true", if not return "false".
df.drop_duplicates()	function	Delete duplicate data in the data of df so that only one row remains for the same rows.
Corrcoef(column1, column2)	function	Calculate the correlation coefficient of two columns of equal row in MATLAB

Table1: the function/variable that will be used in Data preprocessing

2.1.1 Detecting and deletion of duplicate data:

Detection:

We use the df.duplicated() function to determine if there are duplicate lines in all of the lines of training data , and output the detection result.

The output is as follows:

```
The number of original duplicated datas is 164
The rows number of original datas is 30904
```

Deletion:

I use the function “df.drop_duplicates()” to delete duplicate data.

2.1.2 Correlation between attribute and label

I used MATLAB's built-in function “Corrcoef” to calculate the correlation between each

attribute column and the label column in the training set. The results are figure2.

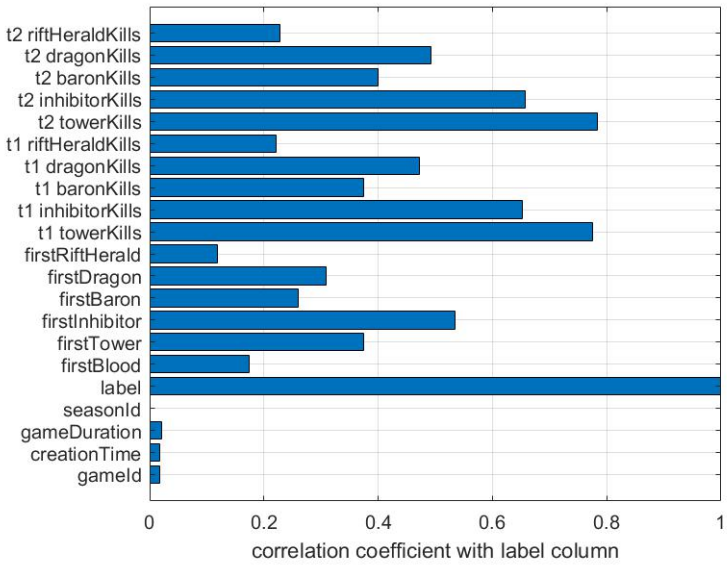

Figure1: correlation coefficient with label column

According to the figure2, we can clearly see that the correlation coefficient between four attributes of the sample "gameId", "creationTime", "gameDuration", "seasonId" and the label are extremely small relative to other attributes. Therefore, I decided to delete them. Now, In addition to the label attribute, the sample has only 16 attributes left.

2.1.3 Max-Min standardization:

In order to narrow the range gap between different attribute values and avoid the impact of attribute value gaps on the evaluation results, I used the **Max-Min standardization** method to normalize the data so that the range of each attribute value of the sample is 0 to 1.

Implementation principle: Assuming that a specific attribute value of a sample is **x**, the maximum value of the attribute corresponding to the entire sample set is **max**, and the corresponding minimum value of the attribute is **min**, then the normalized result **x'** of x can be calculated by the following formula:

$$x' = \frac{x - \min}{\max - \min}$$

2.2 Classification algorithm

2.2.1 Decision tree

2.2.1.1 Principle:

The decision tree can classify the entire feature space step by step by discovering the classification rules contained in the data to distinguish different classification samples.

（1）Training principle:

Decision tree learning is essentially to summarize a set of classification rules from the training data set. We set the judgment condition at each node according to one or more attribute values of the sample, which can divide the training set into multiple branches. The judgment condition set therein should make each sample branch after being divided have a smaller Gini coefficient.

Gini index calculation method:

$$Gini\ Index = 1 - \sum_{i=0}^{c-1} p_i(t)^2$$

symbol	meaning
C	C is the total number of classes
$P_i(t)$	is the frequency of class C at node t
t	Is the number of node

Table2: The meaning of symbol in Gini index

The algorithm repeats the loop process of **"node feature selection-data segmentation-Gini index comparison-node feature selection"** until most branches of the training sample have the correct label. When we have completed the correct segmentation of the training samples, our decision tree is formed.

（2）overfitting

Introduction to over-fitting: If the depth of the decision tree is too large and the nodes are too subdivided, which makes the decision tree highly fit the training set, the generalization ability of the decision tree is poor at this time which means the decision tree cannot make good predictions for unknown samples.

How to avoid over-fitting: Limit the maximum depth of the decision tree to a suitable range. Add the step of "decision tree pruning" during training, and delete those branches that are too fine. Therefore, our training process becomes an iterative process of "node feature selection-data segmentation-Gini index comparison-decision tree pruning-node feature

selection".

（3）Prediction principle:

When the algorithm iteratively uses the process of "feature selection-decision tree generation-and decision tree pruning" to generate the decision tree we want, you can put in any unlabeled sample for category judgment. This sample will make a conditional judgment at each node based on its own characteristics, and be classified into a branch, leaving only a clear category in the end.

2.2.1.2 Parameters

Parameters	Meaning of feature	Choice of Parameters
criterion	The selection method of node feature	Gini index
max_depth	Maximum depth of decision tree	20
class_weight	The weight of each category	same
min_samples_split	Minimum number of samples required for node subdivision	2
min_samples_leaf	The minimum number of samples required to transform a node into a leaf (to avoid overfitting)	1
min_impurity_split	Threshold that determines whether the node becomes leaf based on criterion	0

Table3: The parameters of decision tree

2.2.2 Artificial neural network (ANN)

2.2.2.1 Basic principle introduction:

(1) Introduction of neurons:

In cranial nerves, a neuron has three parts: "dendrites - cell bodies - axons". The dendrites are responsible for receiving information that comes from the input or the axon of another neuron. The cell body is responsible for performing certain functions based on this

information and then generate output information. The axon is responsible for transmitting the output information to the next neuron.

We can implement the function of a neuron in an algorithm, where the dendrite of each neuron is responsible for receiving the parameters of input terminal or the parameters of another neuron′s axon. The cell body performs certain calculations on these parameters, and obtains Output parameters. The axon passes these output parameters to the next neuron.

(2) Introduction of parameter transfer between neural network layers:

In this way, the process of transforming some features into another is similar to mapping. We can achieve multiple mappings of the data set by building a multilayer neural network with multiple neurons in each layer. In this way, complex features can be mapped to simple features, high-dimensional features can be mapped to low-dimensional features, and non-linear separable features can be mapped to linear separable features. The inter-layer relationship trained from the training set with discrete features is an artificial neural network classifier.

(3) training principle:

When each neuron accepts input parameters, it will perform a weighted summation of these parameters, and then directly pass these summed values as output parameters to the next layer, or first substitute into the activation function and then pass the result of function to the next layer until the final output results. When we are training, we will continuously adjust the weight of each neuron according to the difference between the predicted result and the real result, until this multilayer neural network can make good predictions on the training set. The final training result we get is a complete multi-layer neural network and the weight of each neuron.

(4) Prediction principle:

When the neural network is formed, it can take the attribute values of the predicted sample as input parameters, and pass these parameters between layers to finally output an relatively accurate prediction.

2.2.2.2 Introduction to the neural network used in this algorithm:

(1) layers structure

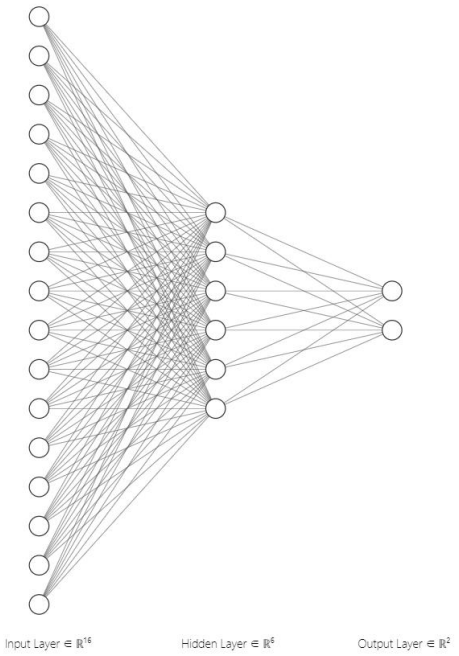

Figure2: Layers structure of ANN

(2)Introduction to layers:

The input layer can perform weighted summation of 16 input feature parameters and using the sigmoid activation function to form 6 new feature parameters.

The Hidden layer can perform a weighted summation on the 6 input features, and then output two feature parameters.

The Output layer will receive the two outputs of the hidden layer with a weight of 1, and then compare the sizes of the two features and output the classification result with the larger corresponding value.

(3) The transfer process of feature vector *X*:

Input -> Weighted summation of the first convolutional layer -> sigmoid function for value conversion -> Enter hidden layer, weighted sum -> Enter output layer, perform size comparison -> Output predicted value.

2.2.3 K-Nearest Neighbor(KNN)

For a given test sample, the KNN algorithm finds the K training samples closest to it in the training set based on the similarity measurement, and then makes predictions based on the information of these K "neighbors". The voting method can be used in the classification task, and the category label that appears most in the K samples is selected as the prediction result.

2.2.3.1 Principle

(1) Training principle:

Since our training set has only 16 attribute values, we only need to choose the "linear scan method〞for classification. At this time, the training process is to create a virtual 16D space and put each sample data into it.

(2) prediction principle

For the "linear scan method", when we put an unknown label sample data into this classifier, the classifier will calculate the similarity between this sample and all the training samples in training set, and then extract K training samples that have the most high similarity. Finally, the majority voting method is used to determine the category of the test sample, that is, choose the category label that appears most in the K samples is selected as the prediction result.

2.2.3.2 parameters

Parameters	Meaning of feature	Choice of Parameters
weights	The weight of similarity of the K neighbor samples	Uniform(weight = 1)
algorithm	Algorithm used by the nearest neighbor method with limited radius	Brute(linear scan)
metric	Method of similarity measure	euclidean

Table4: The parameters of KNN

3.Requirements：

Name of prerequisite packages	funciton
numpy	This is an array math function package. There are

	a large number of array operations inside, and it can also convert other data into array types.
sklearn	This is a library for mechanical learning. I called out the classifier I want to use through this library, and can also perform data preprocessing.
pandas	Through this package, I read the table data type of csv in on the computer.
time	This package can display many times including local time and current system time. I used it for timing.
torch	An open source Python machine learning library. I used its functions to build my multilayer neural network.

Table5: The name and simple introduction of prerequisite packages I used

4.Results：

For the purpose of avoiding occasionality of outcome of training, I set up **10 loops** to train 10 separate classifiers and separately get their accuracy and training time, and finally use the average value of accuracy and training time as the standard of assessing of the classifier.

4.1 Decision tree (DT)

4.1.1 The figure of trained decision tree：

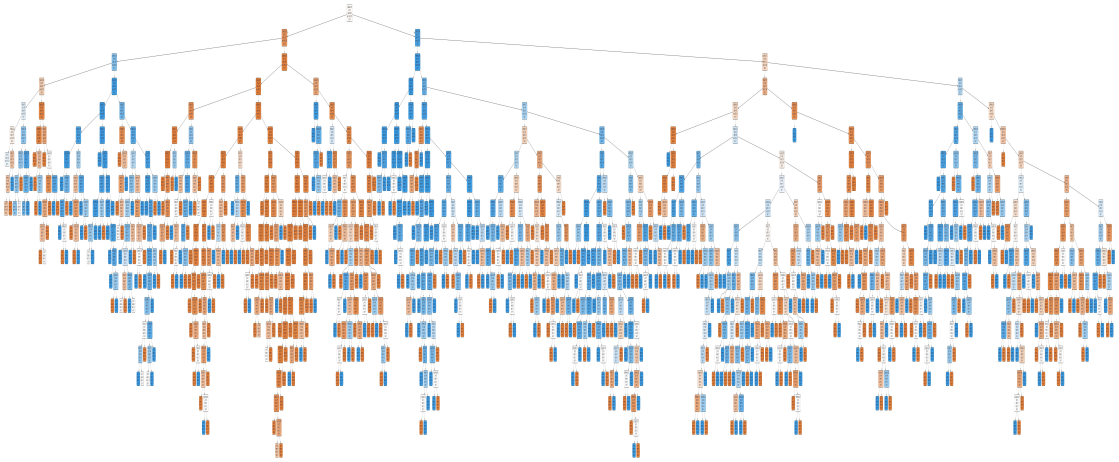

Figure3: Trained decision tree:

4.1.2 Training time and accuracy

Loop 次数	训练时间(s)	准确率
1	0.087	0.9624
2	0.128	0.9633
3	0.089	0.9636
4	0.091	0.9632
5	0.091	0.9638
6	0.091	0.9630
7	0.078	0.9634
8	0.082	0.9636
9	0.077	0.9629
10	0.086	0.9628
Average	0.090	0.9632

Table6: Training time and accuracy of 10 loops of DT

4.2 ANN

Loop 次数	训练时间(s)	准确率
1	2.214	0.9374
2	2.166	0.9525
3	2.170	0.9553

4	2.243	0.9578
5	2.254	0.9590
6	2.536	0.9585
7	2.236	0.9596
8	2.267	0.9600
9	2.237	0.9602
10	2.340	0.9604
Average	2.266	0.9561

Table7: Training time and accuracy of 10 loops of ANN

4.3 KNN

Loop 次数	训练时间(s)	准确率
1	0.853	0.9562
2	0.945	0.9562
3	1.070	0.9562
4	0.918	0.9562
5	0.855	0.9562
6	1.004	0.9562
7	1.015	0.9562
8	1.024	0.9562
9	0.841	0.9562
10	0.770	0.9562
Average	0.929	0.9562

Table8: Training time and accuracy of 10 loops of KNN

5.Comparison and discussion:

5.1 Results of the three classifier algorithms:

We use the average of 10 loop results as the result of this classification algorithm.

Type	Training time	Accuracy
DT	0.090	0.9632
ANN	2.266	0.9561
KNN	0.929	0.9562

Table9:Results of the three classifier algorithms

You can see from the training results that the DT algorithm has the shortest training time and the highest accuracy. The training time of the ANN algorithm is longer, and the training time of the KNN algorithm is shorter.

5.2 Discussion of the quality of each classifier based on this data set:

5.2.1 DT

5.2.1.1Main advantages and disadvantages:

The main advantage is that the model is readable, the classification speed is fast, and it can make feasible and effective results for large data sources in a relatively short time.

The main disadvantages: it is difficult to predict continuous attribute, and continuous data needs to be segmented. When there are too many categories, errors may increase faster. For data with time sequence, the data preprocessing work is required.

5.2.1.2Analysis based on the training set:

1. Since the LOL game information sample set we use does not have continuous features, the algorithm does not need to segment continuous attributes. 2. There are only two sample categories, and this task belongs to a classification task with fewer categories. 3. The chronological attributes was deleted in data preprocessing.

5.2.2ANN

5.2.2.1Main advantages and disadvantages:

Advantages: It can classify samples with a large number of complex non-linear feature by multiple mapping. It often performs better in classification problems with complex features.

Disadvantages: Due to the use of many parameters, the training is relatively

time-consuming. A lot of training data is needed.

5.2.2.2Analysis based on the training set:

This training set only has a few linearly separable feature values, and the advantages of ANN are not shown. The sample size is large, so a training loop requires more calculations, so the training time is longer.

5.2.3 KNN

5.2.3.1Main advantages and disadvantages:

Advantages: simple thinking and easy to understand. The time complexity of training is relatively low, only O(n). Not sensitive to abnormal points. It is suitable for automatic classification of class domains with relatively large sample size.

Disadvantages: The amount of calculation is large, especially when the number of features is very large. When the sample is unbalanced, the prediction accuracy of rare categories is low. The prediction speed is slow.

5.2.3.2Analysis based on the training set:

After preprocessing, the training set has only 16 sample attributes, and the calculation difficulty is relatively low. The sample classification is relatively balanced, and there is no rare class. Therefore, the KNN classification algorithm can maintain a high accuracy rate.

5.3 Comparison among the three classifiers:

In this training sample, the sample has only 16 discrete feature values and only two categories. The advantages of the ANN algorithm are difficult to shown, and due to the larger capacity of the sample, the training time of the ANN algorithm is longer. Choosing a decision tree algorithm is a good choice, because the algorithm can achieve better classification of samples with a small number of categories and a small number of discrete features. Of course, the KNN algorithm can also be selected for classification, because for training samples with fewer attribute values and no rare classes, the KNN algorithm can also make better predictions, but the KNN algorithm requires a longer prediction time.

6.Summary:

The classification algorithm in supervised learning is an effective way to predict the

category of unknown samples. The sample data used in this project comes from the game information of LOL(League of Legends). The purpose of the project is to predict the winner of the game based on the information of the game. Initially, the sample has 20 features and one label. In the process of data preprocessing, after correlation analysis, some of the features of the sample did not have much effect on label prediction, so these invalid features were deleted in the process of data preprocessing, and the sample left 16 attribute features and one label feature . Subsequently, three classifiers (DT, ANN, KNN) were trained using the training set, and then they become the relatively accurate classification model, and they can do a relatively accurate prediction on the test samples.

The next step in this project is to try real-time monitor the information of game, and give game suggestions to players. When a player is playing a game, the game information will be collected by this classifier in real time. This classifier predicts the game result based on the game information, and proposes game suggestions to both parties based on the prediction result and the current game information.