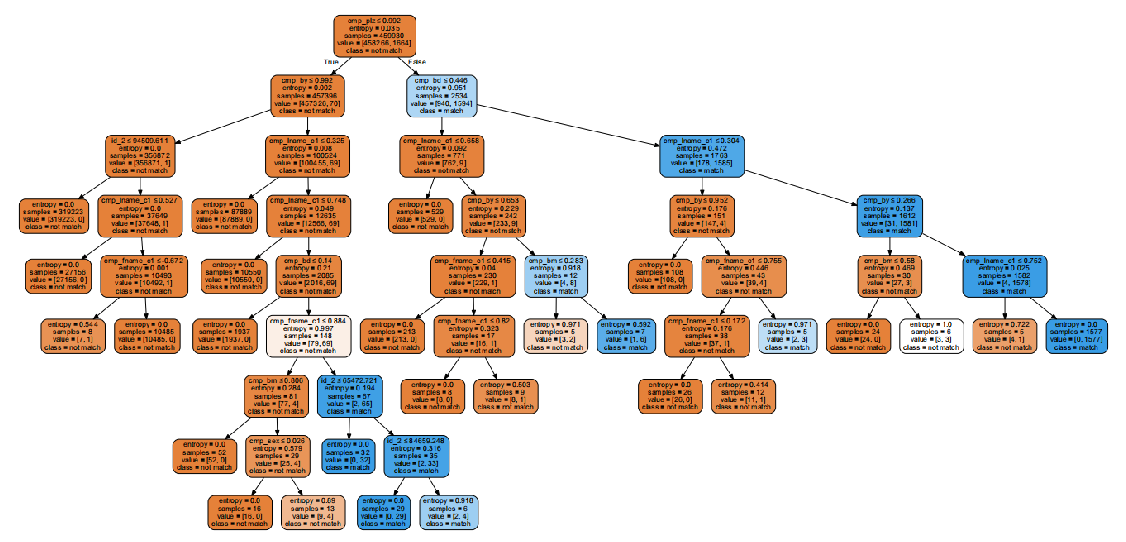
Classification Number Confidentiality level

UDC

“Big Data”Course Final Thesis



**Classification of record link pattern matching Data Set using improved information entropy decision tree algorithm C4.5**

Author Name : Zi Liang

Student ID : 20163933

College :College of Information Science and Engineering

Speciality ： automation(Cybernetics)

Northeastern University

2018 11

**Classification of record link pattern matching Data Set using improved information entropy decision tree algorithm C4.5**[[1]](#footnote-0)

**Abstract**

Decision tree algorithm is one of algorithms of machine learning. It uses attribute values of data to generate tree structure, and thus represents a mapping of object attributes to object values. Each node in the tree represents the judgment condition of object attributes, its branch represents the object that meets the node condition, and the leaf node represents the prediction result that the object belongs to. Dimensionally, decision tree algorithm is mainly used for target task prediction and classification.

This paper chooses the cancer patient identity data set from the Medical Center of Johannesburg University in Mainz, Germany as the target classification data set[[2]](#footnote-1). According to the requirements of Dr. Li, the classification prediction model is built based on the supervised learning decision tree prediction algorithm C4.5 in data mining. Based on this task, the following work has been accomplished:

1) The default (or missing) values of attributes of the data are pre-processed by means of high-dimensional mapping, so that all the information of the original data is preserved without introducing artificial noise.

2) In the use of data sets, all 574 9132 sets of data are divided into ten groups according to the same frequency of matching or not, and the robustness and generalization ability of the algorithm are verified by various proportions of training set, test set and verification set. The accuracy of the algorithm is 99.9956%, 99.9810%, 99.9977% in test set respectively. The problems of under-fitting and over-fitting and the optimal data scale has been analyzed.

3) With decision tree algorithm C4.5, the pessimistic error pruning method, cross-complex pruning method and other methods are used for pruning.

The program is written with Python 3. Pandas Library is used to read target data, numpy Library is used to process default values, scikit-learning Library is used to generate decision tree and divide data sets, graphviz Library and Matplotlib Library are used to visualize data. The source code was written with reference to the source code of the author of C4.5. Thanks all here.

Experiments show that the accuracy of this model in solving this problem is stable at 99.99%. It can be said that the model is completely competent for the task classification and prediction of the data set.

**Keywords:** decision tree ; C4.5 algorithm; information entropy; missing data handling; Pruning method

、

# 目录

Abstract II

目录 IV

1. Decision Tree algorithm C4.5 5

1.1. Iterative Dichotomy 3 Algorithm 5

1.1.1. Information Entropy Theory and Information Gain 5

1.1.2. Specific flow chart of ID3 algorithm 6

1.1.3. Defects of ID3 algorithm 6

1.2. Based on ID3——C4.5 decision tree algorithm 6

1.2.1. Discrete Processing of Continuous Attribute Values 7

1.2.2. Information Gain Rate 7

1.2.3. Missing value handling 8

1.2.4. Pruning method: Error-based Pruning 8

2. Based on C4.5——Custom improvements 8

2.1. Data Pre-processing——High Dimensional Mapping of Attribute Values 8

2.2. defending over-fitting——Pruning strategy compared with CART algorithm 10

2.2.1. PEP: Pessimistic Error Pruning 11

2.2.2. CP: Cost-complexity Pruning 11

2.3. Solving Data Contradictions 13

3. Experimental verification and result analysis 13

3.1. Data set 13

3.2. Experimental verification and result analysis 15

4. Conclusion 19

4.1. Conclusion 19

4.2. Deficiencies 19

5. reference 19

# 

# Decision Tree algorithm C4.5

## Iterative Dichotomy 3 Algorithm[[3]](#endnote-0)

The essence of decision tree algorithm is to classify data according to the attributes of existing data. Therefore, all decision tree algorithms will face the problem of how to select which attribute to make conditional selection.

In order to solve the above problems, Quinlan proposed the Iterative Binary Tree III (ID3) algorithm in 1970. This algorithm is based on Occam's razor theory, that is, the smaller the decision tree, the better of the decision tree. This algorithm belongs to heuristic algorithm, and uses the entropy of Shannon's information theory to measure the decision-making process of decision tree, thus realizing simple and efficient decision-making.

### Information Entropy Theory and Information Gain[[4]](#endnote-1)

The concept of entropy comes from physics. In 1948, Claude Elwood Shannon introduced the entropy of thermodynamics into information theory. According to Boltzmann's H-theorem, Shannon defined the entropy H of a stochastic process X as:



Where P is the probability mass function of random process X, E is the mathematical expectation function,  is the Self information quantity of X.

When taken from a limited number of discrete samples, the entropy of a stochastic process can be expressed as



whereis the logarithmic bottom, which usually takes 2, and so the unit of entropy is bit。

According to the entropy formula of single variable, we can transition to the joint entropy expression of multi-variable. The joint entropy expression of X and Y variables is given below.



Based on the above joint entropy and conditional probability, Conditional entropycan be shown as



According to the definition of conditional probability, all of the stochastic process with multiple variables can be expressed with , and we can use information entropyto express the uncertainty of X when Y is determined, we call it as



whererepresents the improvement of the X after Y is determined, which called as information gain.

ID3 algorithm uses information gain as the criterion to decide which attribute of data should be used by the current node. The larger the information gain, the more suitable the attribute is for classification at present.

### Specific flow chart of ID3 algorithm

Let there be m training samples, and the number of output set is D. Each sample has N discrete attributes. The set of attributes is marked as A and the output is decision tree T. Then the flow of ID3 algorithm can be summed up as follows:

Computing Information Entropy of Output Set D: ;

Determine whether n is greater than 2, if not, the program ends and returns to decision tree T. If n is greater than 2, proceed to step 3.

Computing Conditional Information Entropy of All Attributes in Set A, called , and then compute all information gain of attributes in set A,called , then extract the largest number as the depending to split the decision tree in that node，and then generate the decision tree T, at the same time that attribute will be removed from the set A, then reduce the variable n of the number of recorded attributes by one, and finally jump back to step 2.

### Defects of ID3 algorithm

ID3 algorithm gives a measure method of choosing decision attributes. It solves the problem of choosing attributes as decision tree splitting nodes for the first time, but it still has many shortcomings, including the following aspects:

1) ID3 algorithm does not give a solution for attributes with continuous values, which limits its application.

2) ID3 algorithm chooses decision attributes in the way of maximizing information gain, and often contains attributes with more attribute values, whose information gain is always larger, but their uncertainty is often the same or other results.

3) ID3 algorithm does not provide solutions for default values.

4) ID3 algorithm has not considered the problem of over-fitting.

## Based on ID3——C4.5 decision tree algorithm[[5]](#endnote-2)

To solve the above four problems, Ross Quinlan proposed the C4.5 decision tree algorithm in 1992 . On the basis of ID3 algorithm, this algorithm uses the method of discretizing continuous value attributes to solve the problem of data classification with continuous value attributes[[6]](#endnote-3)，uses the information gain ratio replace the information gain to improve the influence of the number of optional attribute values on the information uncertainty relationship. The idea of dealing with default values is given, and a pruning scheme is proposed to deal with the over-fitting problem. These four aspects are discussed in detail as follows.

### Discrete Processing of Continuous Attribute Values

For the classification task of M samples, if attribute A is the feature of continuous change of attribute values, M-1 partition points can be obtained by calculating the average value of adjacent samples in M samples as partition points. The first partition point can be expressed as, So we can get the idea of generating binary tree according to the M-1 partition point.It is still to calculate the system information gain when different partition points are used as the partition basis, and select the partition point with the greatest information gain as the current decision branch point.

Unlike the branches of discrete values, discrete continuous attribute values do not disappear after being used as decision branches, but continue to be used as a basis for judging binary trees and compared with the information gains of other attributes until all sample attributes under this attribute have the same values.

### Information Gain Rate

Information gain has the characteristic of biasing toward attributes with many attributes. In order to correct this problem, the concept of characteristic entropy is introduced at first.

For a stochastic process whose set of output attributes is D, the characteristic entropy of one of the attributes A is defined as



Where n is the number of attribute values of attribute A,is the number of samples corresponding to the first value of attribute A.

By introducing the characteristic entropy, the information gain can be corrected and the information gain rate can be obtained.



By this way, the selection of decision nodes will be more objective and scientific.

### Missing value handling

The solution of default value in C4.5 is to add the weight reduction of default data into the calculation of information entropy, and then treat default data as data after several decisions.

According to the C4.5 algorithm, for M samples with output feature set D, the default collection of attribute A is m1, the feature set is D1, the default collection of attribute A is m2, and the feature set is D2. When the conditional information entropy is calculated, the formula can be modified to



Where  is the weighted average value.That is, the proportion of non-missing samples in the overall sample.

After making a decision, if attribute A is selected as the decision node, attribute A will disappear in the following node calculation, so the data of missing attribute A value will be applied to two sub-decision trees equally, in this way, the missing value can be processed.

### Pruning method: Error-based Pruning

Pruning is an indispensable step after the generation of decision tree, which will be described in detail in 2.2. In 1992, when Quinlan proposed the C4.5 algorithm, he also gave a pruning strategy: error-based pruning.

This strategy belongs to post-pruning method, that is to say, for each node, the number of false judgements on the verification set before and after pruning should be calculated. If pruning helps to reduce false judgements (including equal cases), the branch where the node is located should be deleted, and vice versa.

This scheme belongs to the method of dealing with moderate rules and regulations. This paper improves it in practical use.

# Based on C4.5——Custom improvements

## Data Pre-processing——High Dimensional Mapping of Attribute Values

While handling the data set Record Linkage Comparison Patterns,I found there are a number of missing number, which is repeated occurred in Record,just like figure 2.1 show.

Decision tree algorithm always has poor stability. Because of this, the change of attribute values may cause fundamental changes in the structure of decision tree, so it is necessary to deal with these missing attribute values reasonably.。We have talk about it in chapter 1.2.3 that C4.5 algorithm see having missing number or not as weight. However,they take back the data after decision.In this way, the missing attribute values are simulated and predicted, but there are many problems. First, we don't know what the values of the attributes are in the actual data. Therefore, it is dangerous to use the known attribute values to infer the unknown attribute values, so that the local features can be regarded as global features. It makes the final prediction result deviate. In addition, by adding the same part of the default attribute sample set to the sub-decision tree, the size of the decision tree as a whole will be increased, and the computational load will be increased, and other problems such as over-fitting will easily arise.

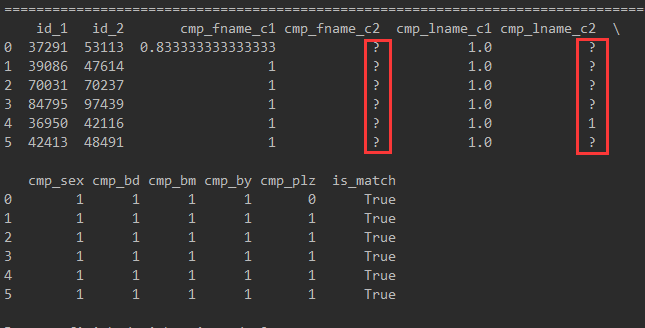


figure2.1 the first sixth records of the data set Record Linkage Comparison Patterns,which the number in red box is missing

To solve the above problems, this paper does not use the strategy of weighted information entropy, but maps the attribute values of each attribute to high-dimensional space, that is, the default value is also treated as an attribute value. With this scheme, the default value will represent the same degree of a feature, thus realizing the non-destructive correction of the data and eliminating the prediction link of the default value. The advantage of this method is to ensure the original use of data. The disadvantage is that the default data may come from different attribute values under the same attribute, but this source can not be represented in the data, so in theory, this method can achieve the best use of data conservatively.

Each missing attribute value under the same attribute is replaced by the same coincidence, and then the symbol is used as an attribute value in the decision-making, thus realizing the meaning of retaining the default attribute value without changing the original meaning of the data. The relevant processing code and effect are shown in Figure 2.2.

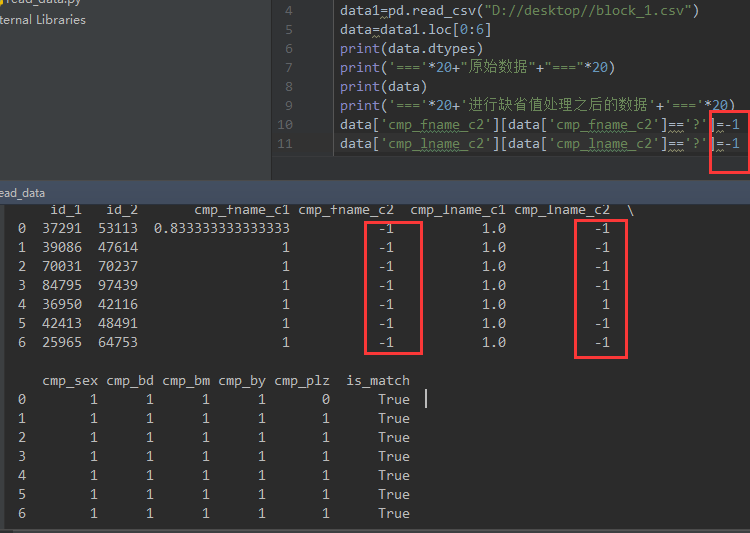


figure2.2 processing to missing value

## defending over-fitting——Pruning strategy compared with CART algorithm

Over-fitting means that the error of the model in the training set is very small, but very large in the validation set. Usually the over-fitting problem is caused by the much complexity of the model, as shown in Figure 2.3. Therefore, in order to restrain the rapid expansion of the model, some specific measures must be taken. Decision tree algorithm is prone to over-fitting problem because of its many parameters, so appropriate measures must be taken to deal with it.

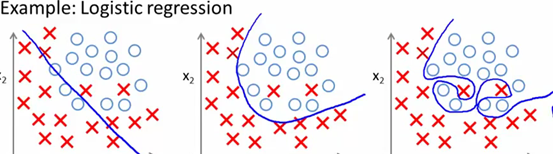


figure2.3 under fitting,fitting,and over fitting

Generally, decision tree algorithms such as ID3 and C4.5 will reduce the size of decision tree as much as possible, which is also an important factor for the introduction of information entropy. However, the saturation problem of decision tree algorithm is still unsatisfactory. So almost all decision tree algorithms need pruning.

At present, the main strategies of decision tree pruning are: reduced error pruning, pessimistic error pruning, minimum error pruning, cost-complexity pruning, error-based pruning and critical pruning.

Since error-based pruning has been introduced in the previous section, the following two methods are mainly discussed: pessimistic error pruning and cost-complexity pruning.

### PEP: Pessimistic Error Pruning

PEP pruning belongs to top-down pruning. Since only training set errors are used in the construction process, no additional verification set is required. At the same time, 0.5 is often added as a correction factor in order to improve the error caused by the bias of pruning process to training data.

The error rate of a tree with T nodes is measured as



is the number of error predictions at node t, N(t)is the number of all predictions at node t.

For above decision tree, after removing node K,the error rate of node is



If we make,then we have



That is to say, the change of error rate after removing a node is equal to the total number of nodes under the node after subtracting the number of false samples before removing them.

So the pruning condition can be expressed as



### CP: Cost-complexity Pruning

The scheme adopted in this paper is the cost-complex pruning method, which first appeared in the CART decision tree algorithm proposed by Breiman in 1984.

The idea of this pruning method is to generate decision trees first, then generate all possible pruned decision trees, use cross validation to test the effect of various pruning, and finally select the pruning strategy with the best generalization ability. So pruning method can be divided into two steps. The first step is to generate all kinds of pruned decision trees from the original decision tree. The second step is to use cross validation to detect the prediction ability after pruning, and select the pruned tree with the best generalization prediction ability as the final decision tree.

So there are two problems to be solved here: one is how to prune, the other is how to evaluate the generalization prediction ability of decision tree after pruning. We need define the cost function first.

The pruning idea of the cost-complex pruning method comes from the regularized linear regression operation in linear regression.

The basic expression for regularization of linear regression is



whereis hyper-parameter，and we can changeto change the order of polynomials.

Similarly, the CP pruning method uses a hyper-parameter to control the depth and breadth of the decision tree. In CART algorithm, the loss function measure of pruning is defined as



whereis the hyper-parameter which is the same as, andis the train error of the decision tree, which using GINI ratio in CART:



In C4.5 algorithm, Shannon information entropy formula is still used for training error loss function.is the number of leaf in the sub-tree.

In this way, the definition of pruning loss function is completed.

It can be seen that, similar to the limitation of higher order number in linear regression, CP pruning method restricts the breadth of pruning.Therefore, the size of decision tree can be improved by adjusting the size of the super-parameters,，the largeris， the smaller the decision tree is.

Using this metric to prune, the thresholds of whether or not each sub-decision tree is pruned can be calculated as



For each to cross validation，we can find the best value of, we can use it to find the best sub-tree.

The above steps are summarized as follows:

Initialize，the best sub-tree set；

1. Compute the train error cost functionfrom the leaf node to the node T,then compute number of leaf,while writing down the ratio of the normalization,update；
2. get the set of the nomal parameters of all the node M;
3. Select the largest value from M,then access the internal nodes of subtree t from top to bottom,if,then prone. After that we wil get the best sub-tree of；
4. Add the obtained optimal subtree to the set of optimal subtrees,removefrom M；
5. Run it until M is none；
6. Selecting the Optimal Subtree in the Optimal Subtree Set by Cross-validation.

## Solving Data Contradictions

The problem of data contradiction refers to the phenomenon that the attribute values of two sets of data with the same attributes are completely contradictory. Decision tree algorithm can not borrow such data to generate decision tree model. Of course, after the decision tree model is generated, it can be distinguished by the data after pruning. Generally, the maximum matching principle is adopted, that is, the maximum number of different results on leaf nodes is selected as the prediction result. However, this ambiguous leaf node will bring difficulties to the calculation of pruning process when it is generated, so this paper proposes a virtual leaf node method.

Virtual leaf nodes refer to the problem of contradiction existing at the beginning of decision tree generation, which is solved by adding a new layer of leaf nodes to the contradictory leaf nodes. When the leaf node is visited, the number of nodes is considered to be the number of all samples involved. The predicted value of the node is based on the number of attributes of the samples in different results, and the final predicted value is obtained by means of random probability.

In this way, not only the contradictions widely existing in the data are solved, but also the contradictions of the data are reasonably explained, thus taking into account the accuracy of the model and the fidelity of the data in general.

# Experimental verification and result analysis

## Data set

The data set used in this assignment is the Cancer Patient Identity Data Set of Institute for Medical Biostatistics, Epidemiology and Informatics (IMBEI), University Medical Center of Johannes Gutenberg University, Mainz, Germany. The records of the data set are from the same individuals who have been registered many times, and the correlation degree between the characteristics of each registration is used to determine whether the two records originate from the same patient. Therefore, the data set is called Record Linkage Comparison Patterns, which belongs to the sub-problem. Class predictive problem, which accords with the teacher's requirements for processing data sets. The data set is derived from UCI library according to the teacher's requirements.

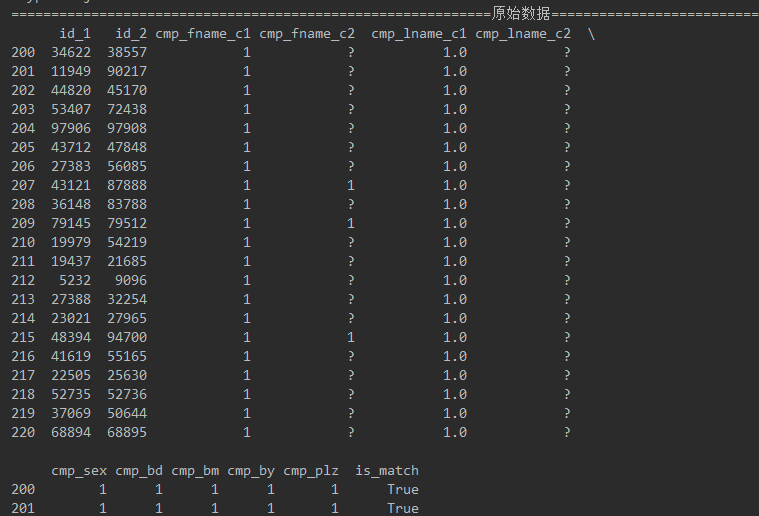


figure3.1original data

According to Figures 3.1 and 3.2, there are 11 input attributes of the data set, representing the account number of the first record, the account number of the second record, the similarity degree of the first part of the surname, the similarity degree of the second part of the surname, the similarity degree of the first part of the name, the similarity degree of the second part of the name, and whether the gender is the same. Whether the date of birth is the same, whether the month of birth is the same, whether the year of birth is the same, whether the postal code is the same, etc. There is only one output attribute, that is, whether the two accounts match.

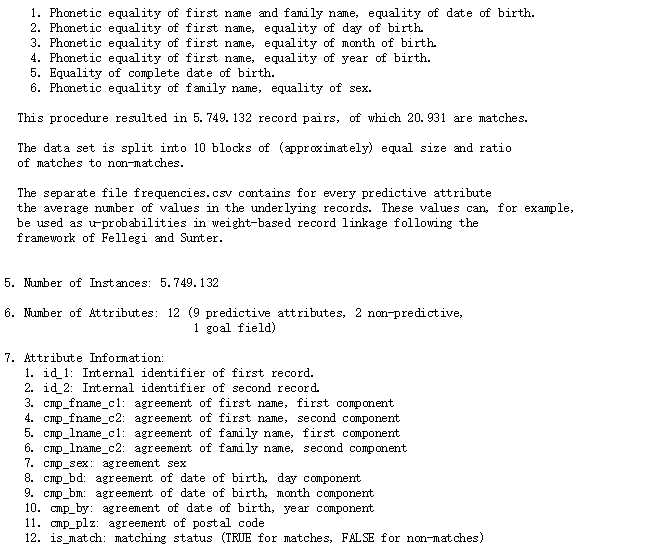


Figure3.2 the explain of the attribute

Among these attributes of input and output, we can find that there are both continuous and discrete Boolean quantities, both fully filled data and default attributes. Therefore, the processing of this data set is not difficult, and there are certain requirements for data processing and overall strategy, which is conducive to the lecture. Use the knowledge and extra knowledge.

In practical application, this assignment adopts the common processing scheme in machine learning, and divides the whole data set into three parts:

Training Set. Used to generate decision tree;

Test Set. Used for pruning to judge whether the model is over-fitting or not.

Validation Set. Used to evaluate the error rate, accuracy and so on.

Moreover, because decision tree algorithm C4.5 is not a large model like deep neural network, the job uses different proportions in the number of data bars of three data sets: I divide all 5749132 groups of data into ten groups according to the same frequency of matching or not, and use training set, test set and verification set separately. The robustness and generalization ability of the algorithm are verified by the ratio of 7 groups, 2 groups, 1 group, 5 groups, 4 groups, 1 group, 1 group, 8 groups and 1 group, and the problems of under-fitting and over-fitting and the optimal data scale are analyzed. The results are shown in 3.2.

## Experimental verification and result analysis

Firstly, the first block of data set is used in this experiment, which is divided into training set and test set. After testing, the decision tree generated is shown in Figure 3.3 and Figure 3.4.

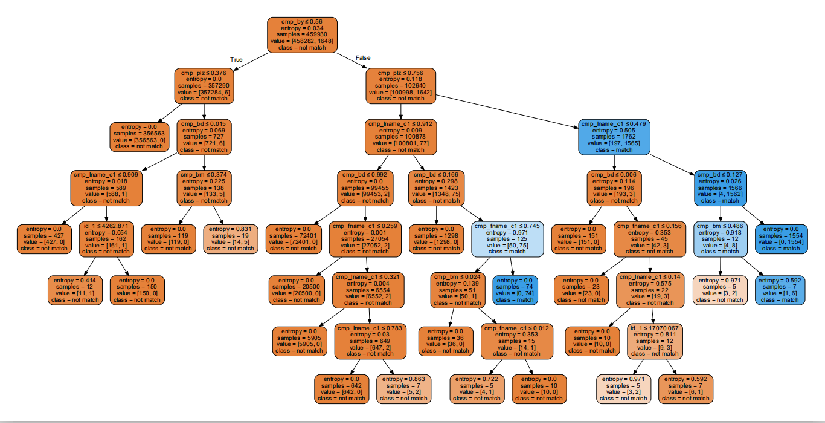


figure3.3 Decision tree sketch (because the decision tree is large, so first put a sketch, the following is a partial enlargement map)

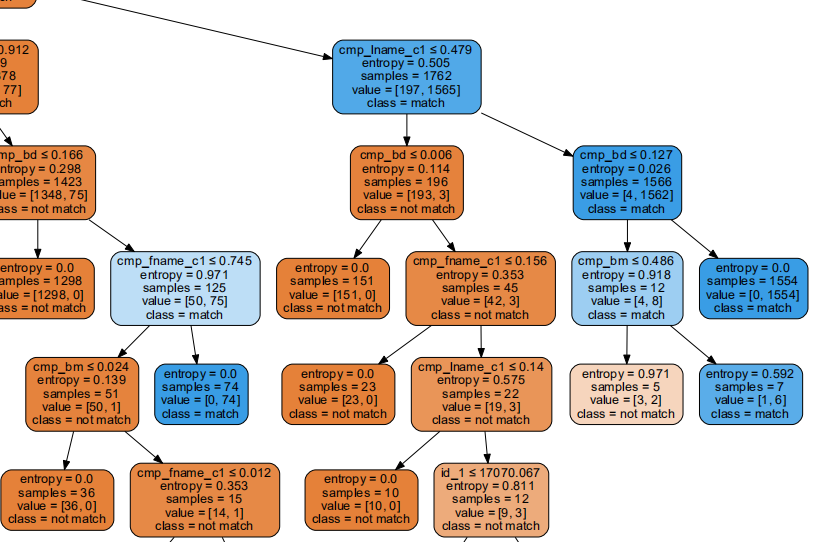


figure3.4 Decision Tree Local Enlargement Map (each box includes split attribute value, information entropy value, sample number, sample distribution under the target attribute, and matching conclusions based on the distribution)

Note that for each node, the left-hand generation represents True and the right-hand generation represents False. The result under the node is represented in the last line. If matched, it is "match" and vice versa, it is "not match". Because I limited the minimum sample number of leaf nodes in the program (the purpose is also to limit the breadth of the decision tree), the minimum sample number of all leaf nodes in the graph is 5.

Using the above decision tree to test the test set, the results shown in Figure 3.5 can be obtained.

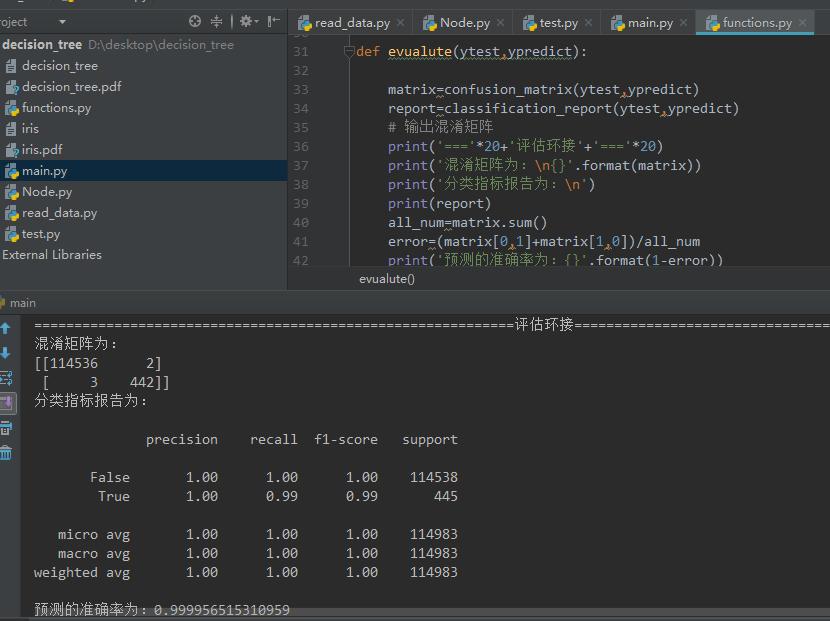


figure3.5 test result

It can be seen that only two of the 110,000 data tested are "mismatched" but marked as "matched". Three of the data are "matched" but are divided into mismatches. It can be seen that even if the decision tree algorithm is used, it is very easy to over-fit and sensitive to the change of attributes. After pretreatment and pruning, the accuracy of the algorithm still reaches 99.99565153%, which is strong enough!

Next, the experiment tried to increase the number of samples again.

When all the samples are used, the decision tree shown in Figure 3.6 can be obtained.

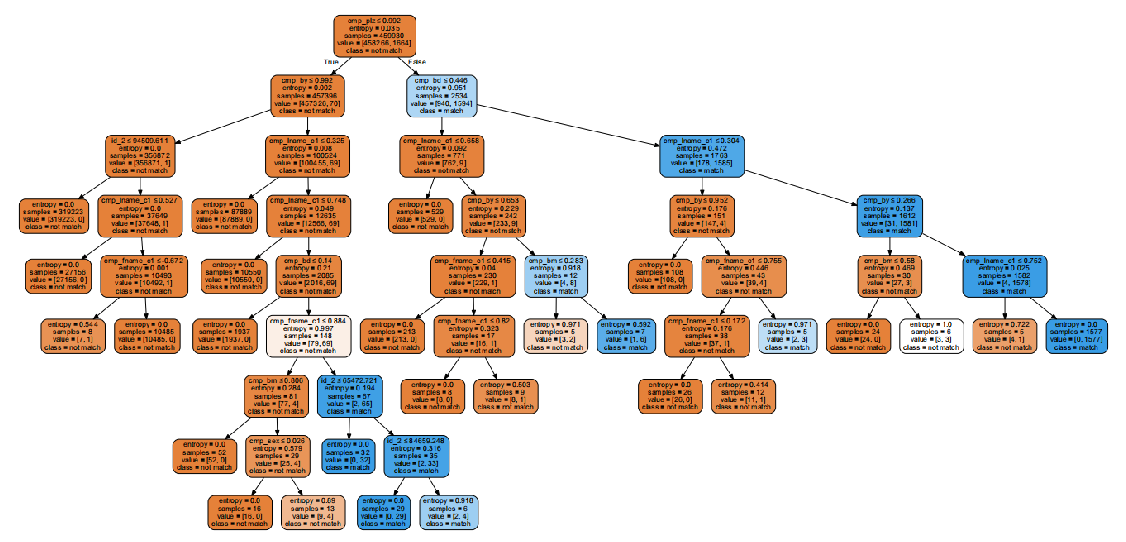


figure3.6 Using 80% of the data for training to generate decision tree model silhouettes

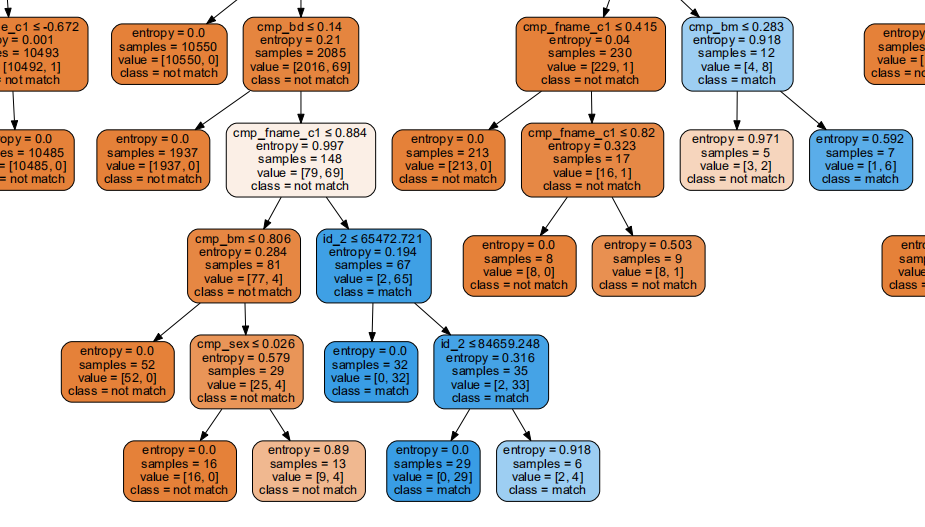


Figure 3.7 part of the decision tree

It can be seen that after training with 80% of the data, the generated decision tree is very similar to the previous decision tree (see Figure 3.4) in the overall structure, but there are some differences in the final location of the leaf nodes. From this, we can see that the decision tree has obvious EXPLANABILITY (white box), which is better than the black box system such as neural network to give a reasonable explanation. The evaluation results of the model are shown in Figure 3.8.

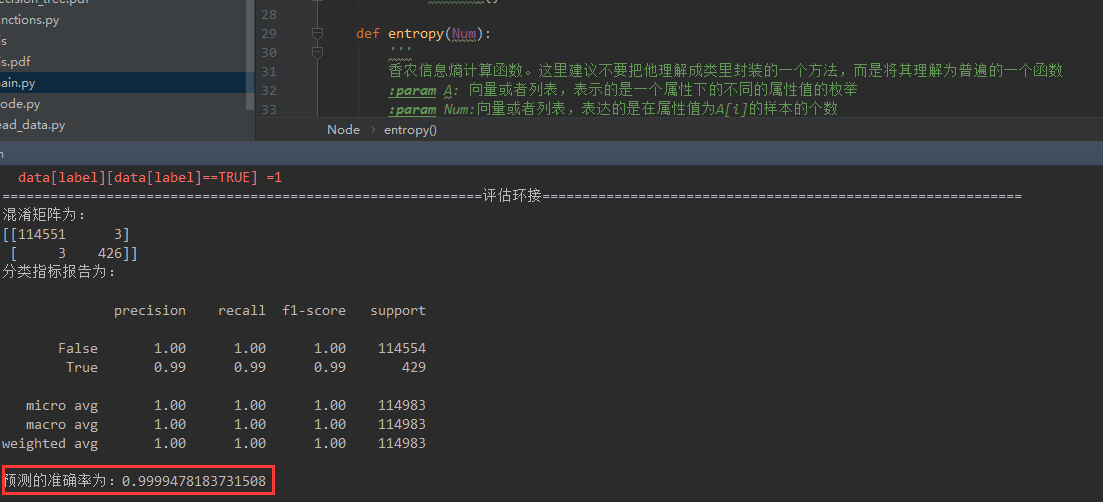


Figure 3.8 run result

As can be seen from the graph, the accuracy of the model on the test set is 99.9947818%, and only 6 samples have classification errors. The accuracy is already very high.

# Conclusion

## Conclusion

After carefully studying each part of C4.5 decision tree algorithm, according to the specific needs of real data sets, this paper determines the most widely used default value processing method mapped to high dimensions by consulting data, which enables the C4.5 algorithm to achieve the processing of data sets. On this basis, this paper analyses and learns several pruning strategies commonly used in decision tree algorithm, and on this basis, chooses one of the most suitable pruning strategies for this data set, and determines the proportion selection of training set and test set after several adjustments.

Finally, after processing, the experimental accuracy of the algorithm in this paper is all over 99.99%. It can be said that the task of two-class classification has been accomplished satisfactorily.

## Deficiencies

When the training sample is biased to a certain feature proportion, the algorithm may perform poorly in practical application. This problem has been corrected by applying a correction factor in this algorithm, but it needs to be adjusted several times.

This algorithm is difficult to deal with complex logic, such as XOR, which is the common fault of all decision tree algorithms. According to the principle of "there is no free lunch in the world", decision tree algorithm has advantages in other places, so it will have disadvantages such as in here. Therefore, it is necessary to analyze the actual classification tasks. If the logic is too complex, replacement algorithms, such as neural networks, can be considered.

# reference

1. See here for the source code : https://github.com/liangzid/DecisionTreeC4.5 [↑](#footnote-ref-0)
2. Here is the website of the data set : http://archive.ics.uci.edu/ml/machine-learning-databases/00210/ [↑](#footnote-ref-1)
3. 《统计学习方法》，李航著。清华大学出版社，2012，3 [↑](#endnote-ref-0)
4. Douglas Robert Stinson; Maura Paterson. 第2.4节“熵”. Cryptography Theory and Practice [密码学理论与实践] [↑](#endnote-ref-1)
5. Quinlan, J. R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993. [↑](#endnote-ref-2)
6. S.B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques", Informatica 31(2007) 249-268, 2007

   In addition, I also refer to a large number of blogs and source code, which is difficult to list one by one. I apologize for that.

   # Appendix

   See the folder attachment for the complete source code, or refer to the website: (https://github.com/liangzid/DecisionTreeC4.5).

   The following is only the source code of the main function:

   import read\_data

   import functions

   from sklearn.model\_selection import train\_test\_split

   import sklearn.tree as tree

   import sklearn

   import graphviz as gvz

   train\_path= ['D://desktop//dataset//block\_1.csv','D://desktop//dataset//block\_2.csv',

   'D://desktop//dataset//block\_3.csv','D://desktop//dataset//block\_4.csv',

   'D://desktop//dataset//block\_5.csv','D://desktop//dataset//block\_6.csv',]

   test\_path = ['D://desktop//dataset//block\_7.csv','D://desktop//dataset//block\_8.csv',

   'D://desktop//dataset//block\_9.csv',]

   vali\_path =['D://desktop//dataset//block\_10.csv',]

   target\_label='is\_match'

   # 读取数据

   DataSet=read\_data.ReadAData(train\_path[0])

   for i in range(9):

   ii=i+2

   path1='D://desktop//dataset//block\_'

   path2='.csv'

   dataset=read\_data.ReadAData(path1+str(ii)+path2)

   DataSet.append(dataset)

   DataSett=functions.fromNaNToLable(DataSet)

   #从数据中提取训练数据和标签

   X=DataSett.drop(target\_label,axis=1)

   Y=DataSett[target\_label]

   #将数据分为训练集和测试集

   Xtrain,Xtest,ytrain,ytest=train\_test\_split(X,Y,test\_size=0.2)

   #print('++++++++++++++++++++++++++++++++++++++++++++++++++++++++')

   #print(DataSett.dtypes)

   #============================================生成决策树================================================

   clf=tree.DecisionTreeClassifier(criterion='entropy', #采用信息熵的计算方式

   splitter='random', #随机在部分划分点中寻找局部最优

   max\_features=None, #最大特征数

   max\_depth=100, #最大深度

   min\_samples\_split=5, #分割所需要的最小样本数

   min\_samples\_leaf=5, #身为叶子所需要的最小样本数

   min\_weight\_fraction\_leaf=0,

   max\_leaf\_nodes=None,

   min\_impurity\_decrease=0,

   #min\_impurity\_split=0

   )

   clf=clf.fit(Xtrain,ytrain)

   #======================================保存决策树=====================================================

   dot\_data=tree.export\_graphviz(clf, out\_file=None,

   feature\_names=DataSet.columns[:-1],

   class\_names=['not match','match'],

   filled=True,rounded=True,

   special\_characters=True)

   graph=gvz.Source(dot\_data)

   graph.render('decision\_tree')

   #进行预测

   predict=clf.predict(Xtest)

   #进行评估

   functions.evualute(ytest,predict) [↑](#endnote-ref-3)