Your name: Zhanghao Kong Course code (e.g. COMP309, COMP440): COMP309

COMP 309 | Machine Learning Tools and Techniques

Assignment 1: Sprint on One Dataset Each
16% of Final Mark | Due: 11:59pm Tuesday 4 August 2020

2.1 Core: Investigate Basic Use of The Different Tribes of AI [50 marks]

Bayesian : Naive Bayes

```
=== Summary ===

Correctly Classified Instances        7785               95.8272 %
Incorrectly Classified Instances       339                4.1728 %
Kappa statistic                          0.9162
Mean absolute error                      0.0419
Root mean squared error                  0.1757
Relative absolute error                  8.3962 %
Root relative squared error             35.1594 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.922    0.008    0.991      0.922    0.955      0.918    0.998     0.998     p
                 0.992    0.078    0.932      0.992    0.961      0.918    0.998     0.998     e
Weighted Avg.    0.958    0.044    0.960      0.958    0.958      0.918    0.998     0.998

=== Confusion Matrix ===

    a    b   <-- classified as
 3609  307 |    a = p
   32 4176 |    b = e
```

Connectionist : Multilayer Perceptron

```
=== Summary ===

Correctly Classified Instances        8044               99.0153 %
Incorrectly Classified Instances        80                0.9847 %
Kappa statistic                          0.9803
Mean absolute error                      0.0209
Root mean squared error                  0.0983
Relative absolute error                  4.1755 %
Root relative squared error             19.6773 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.980    0.983     0.991     p
                 1.000    0.020    0.981      1.000    0.991      0.980    0.983     0.956     e
Weighted Avg.    0.990    0.011    0.990      0.990    0.990      0.980    0.983     0.973

=== Confusion Matrix ===

    a    b   <-- classified as
 3836   80 |    a = p
    0 4208 |    b = e
```

Symbolists : Hoeffding Tree

```
=== Summary ===

Correctly Classified Instances        6652               81.8808 %
Incorrectly Classified Instances      1472               18.1192 %
Kappa statistic                          0.6418
Mean absolute error                      0.1041
Root mean squared error                  0.2396
Relative absolute error                 20.8503 %
Root relative squared error             47.9502 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.999    0.349    0.727      0.999    0.842      0.687    0.986     0.979     p
                 0.651    0.001    0.999      0.651    0.788      0.687    0.986     0.988     e
Weighted Avg.    0.819    0.169    0.868      0.819    0.814      0.687    0.986     0.984

=== Confusion Matrix ===

    a    b   <-- classified as
 3914    2 |    a = p
 1470 2738 |    b = e
```

Analogizers : IBk

```
=== Summary ===

Correctly Classified Instances        8117               99.9138 %
Incorrectly Classified Instances         7                0.0862 %
Kappa statistic                          0.9983
Mean absolute error                      0.0015
Root mean squared error                  0.0276
Relative absolute error                  0.3019 %
Root relative squared error              5.5325 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.998    0.000    1.000      0.998    0.999      0.998    1.000     1.000     p
                 1.000    0.002    0.998      1.000    0.999      0.998    1.000     1.000     e
Weighted Avg.    0.999    0.001    0.999      0.999    0.999      0.998    1.000     1.000

=== Confusion Matrix ===

    a    b   <-- classified as
 3909    7 |    a = p
    0 4208 |    b = e
```

Techniques / results :

	Naive Bayes	Multilayer Perceptron	Hoeffding Tree	IBk
Correct	95.8272	99.0153	81.8808	99.9138
Incorrect	4.1728	0.9847	18.1192	0.0862

In conclusion, IBk have the highest correctly classified instances is 99.9138%

Bayesians : Naive Bayes

General Description :

Naïve Bayes is a probability-based algorithm and it is based on the Bayes's theorem. In the classification process, the algorithm will choose the best result which is with the highest probability. However, an important criterion of success is that all features should be independent with each other.

$P(A|B)= P(B|A)\,P(A)\,/P(B)$

Representation :

Representation of Bayesian can be graphical models. The whole algorithm uses the Bayesian formula to calculate the probability. Therefore, it belongs to the **Bayesian** family.

Evaluation method:

Posterior Probability.

Bayesian can be evaluated by posterior probability , the higher the posterior probability we get the better performance it is. Because the function is unknown, for bayesian it will generate a random function. As we import the training set it will take the evaluations, which are treated as data, the initial function is updated to form the posterior distribution over the objective distribution. Then the posterior distribution will be used to find the next query point.

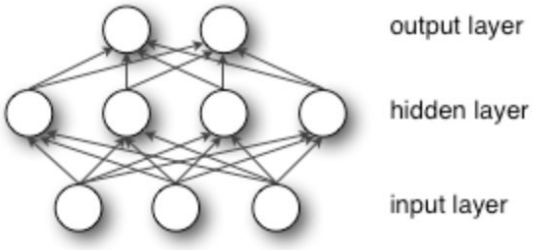
Optimization:

Probabilitic Inference.

Connectionist : Multilayer Perceptron

General Description :

Multilayer Perceptron, In addition to the input layer and the output layer, there can be multiple hidden layers in between. The simplest MLP has only one hidden layer, that is, a three-layer structure, as shown below:



As can be seen from the above figure, the multilayer perceptron layer is fully connected to the other layers.
There is nothing to say about the input layer. For example, if the input is an n-dimensional vector, there are n neurons.
The whole model of MLP is like this.
The three-layer MLP mentioned above is summarized by the formula. The function G is softmax.

$$f(x) = G(b^{(2)} + W^{(2)}(s(b^{(1)} + W^{(1)}x))),$$

Therefore, all parameters of the MLP are the connection weights and offsets between the layers, including W1, b1, W2, and b2.

Representation :

Neural network

Evaluation method:

squared error

Optimization:

To solve the optimization problem, the simplest is gradient descent method: first, all parameters are initialized randomly, then iterative training is carried out, and then the gradient is continuously calculated and the parameters are updated until certain conditions are met. (for example, when the error is small enough and the number of iterations is enough).

Analogizers： IBk(K-nearest neighbours classifier.)

General Description :

K-nearest neighbor (KNN) algorithm is a simple and easy to implement supervised machine learning algorithm, which can be used to solve classification and regression problems. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are similar to each other.

Supervised machine learning algorithm (as opposed to unsupervised machine learning algorithm) is a learning function that relies on labeled input data and generates appropriate output when new unlabeled data is given.

Representation :

The model representation for KNN is the entire training dataset.

Evaluation method:

Large margin nearest neighbor (LMNN)[1] classification

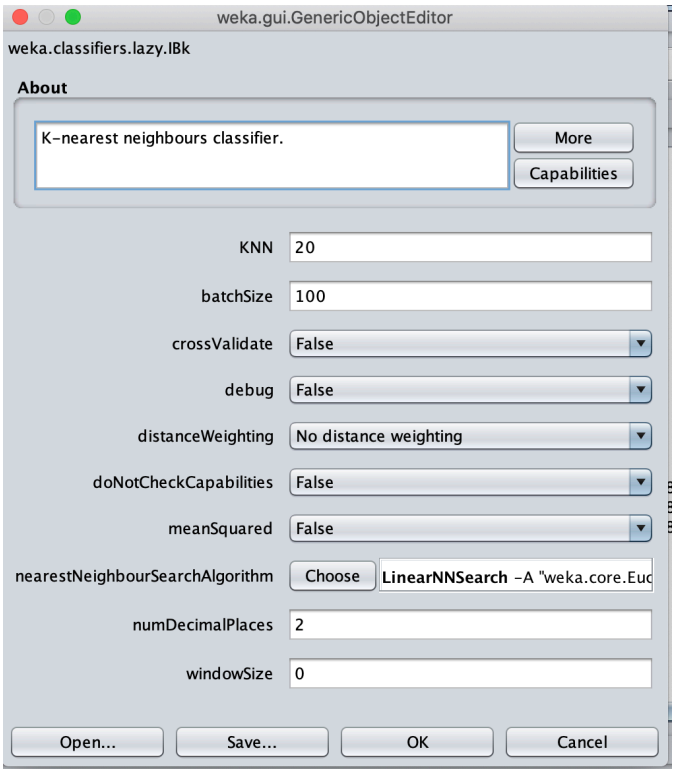
The algorithm doesn't learn a model but it chooses to memorize the instances in the training set. And uses this memory for the prediction phase.Here is one of the popular choice, Euclidean distance is given by :

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + (x_2 - x'_2)^2 + \ldots + (x_n - x'_n)^2}$$

Optimization:

Constrained optimization

Choose a suitable K value for KNN is quite important.When K is small, says 1 we are restraining, and our classifier cannot consider the overall distribution. On the other hand, it will provide the most flexible fit, which will provide low bias but a very large variance. I tried K value from 5 to 30. and When I change the K value to 20 I get the best performance at 99.9138%. Even though I got 100% within 5 to 30, but it will cause overfitting. So I didn't take that value.



Symbolists : Hoeffding Tree

General Description :

A Hoeffding tree (VFDT) is an incremental, anytime decision tree induction algorithm that is capable of learning from massive data streams, assuming that the distribution generating examples does not change over time.

Representation：

problems, logic and search.

Evaluation method:

Accuracy.
What we want to get is pure splitting, that is, splitting into pure nodes, hoping to find an attribute, one of which is "yes" and the other is "no". This is the best case because if it is a hybrid node, it needs to be split again. (my dataset is the best case.)
Quantification is used to determine the attributes that produce the purest child nodes and calculate the purity (the goal is to get the smallest decision tree). The top-down tree induction method uses some heuristic methods -- the heuristic method of generating pure nodes. It is based on information theory, namely information entropy, which measures information in bits.
Information gain = information entropy of distribution before splitting - information entropy of distribution after splitting.

Optimization:

Pruning is an optimization method of decision tree. Pruning is cut from complex subtrees and replaced by a simple tree structure. In the real world, we need to consider not only the accuracy of DT, but also the cost of time. Pruning can help us find the balance point and optimization point of the algorithm.

Differences between each techniques:

In this dataset(Mushroom dataset), Training is used for identifying poisonous and nontoxic mushrooms. Symbolists is used to predict the results of the deduction and inverse deduction of symbols. Connectionist is using neuron network, its characteristics indicate that MP is more suitable for dealing with non-linear correlation dataset. When I am trying to use IBk(KNN) from tribe Analogizers, the correctness is always near 100% or 100%. The reason why it shows the perfect match because overfitting occurs. Maybe it's because I fixed missing attribute.

Finally Naive Bayes(Bayesian) is the most appropriate method I've chosen for this dataset. Bayesian is focusing on subjective probability estimation, occurrence probability correction, optimal decision. For identifying the mushroom, it could be the best way. With correctness 95.8272% which is a very ideal result, it's also a good reason to choose this method.

2.2 Completion 1: Consider a Pipeline for Dataset Processing [20 marks]

Business understanding

Mushroom dataset recorded all attributes which poisonous or nontoxic mushrooms have.We can use Naive bayes algorithm to train then to identify which mushroom is poisonous or not. Agricultural people can use this data to distinguish between poisonous and nontoxic mushrooms, and then grow nontoxic mushrooms for sale. People living in the wild can also distinguish nontoxic mushrooms as food.

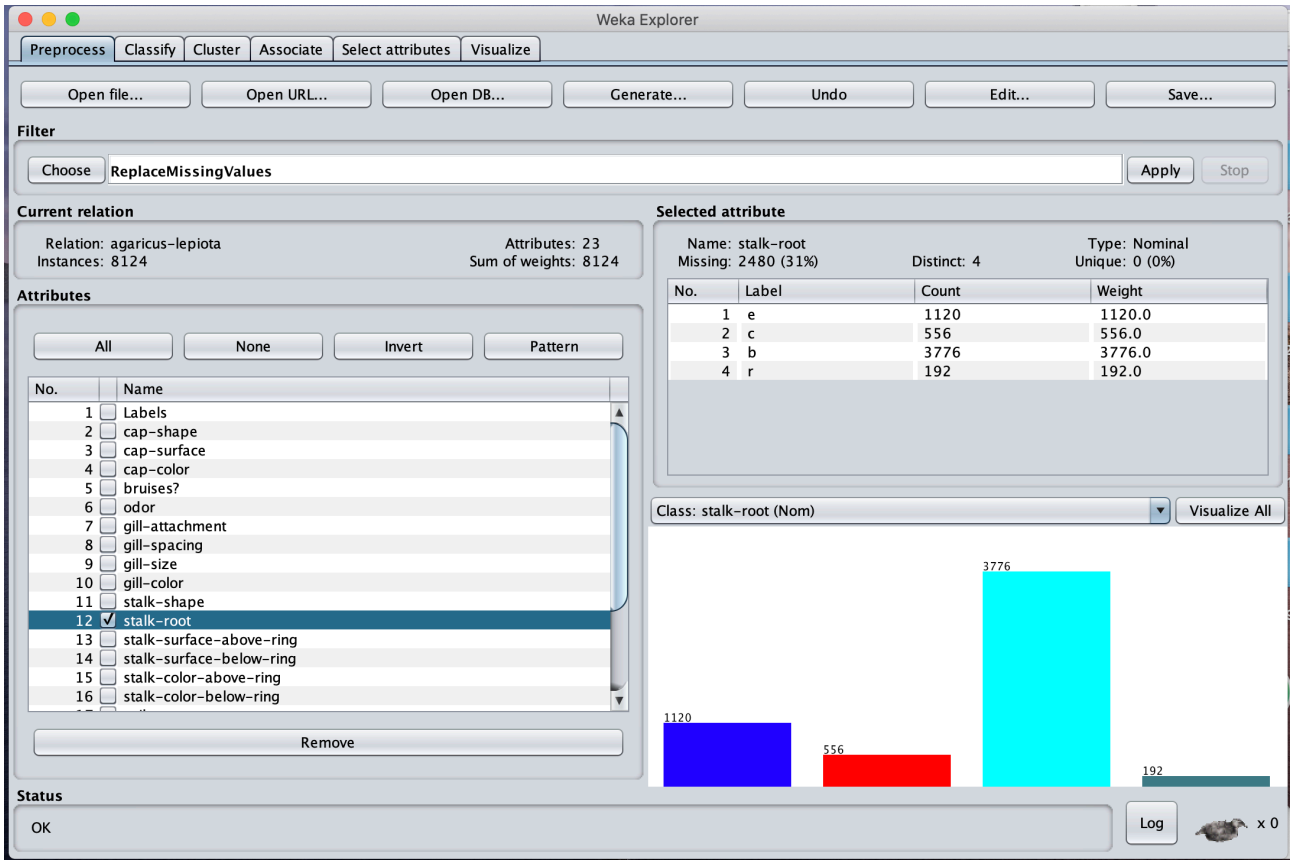
Data understanding

Mushroom dataset recorded 22 different attributes of poisonous or nontoxic mushroom. It has missing attribute for attribute "stalk-root". And the balance of data is also not good enough(edible: 4208, poisonous: 3916).

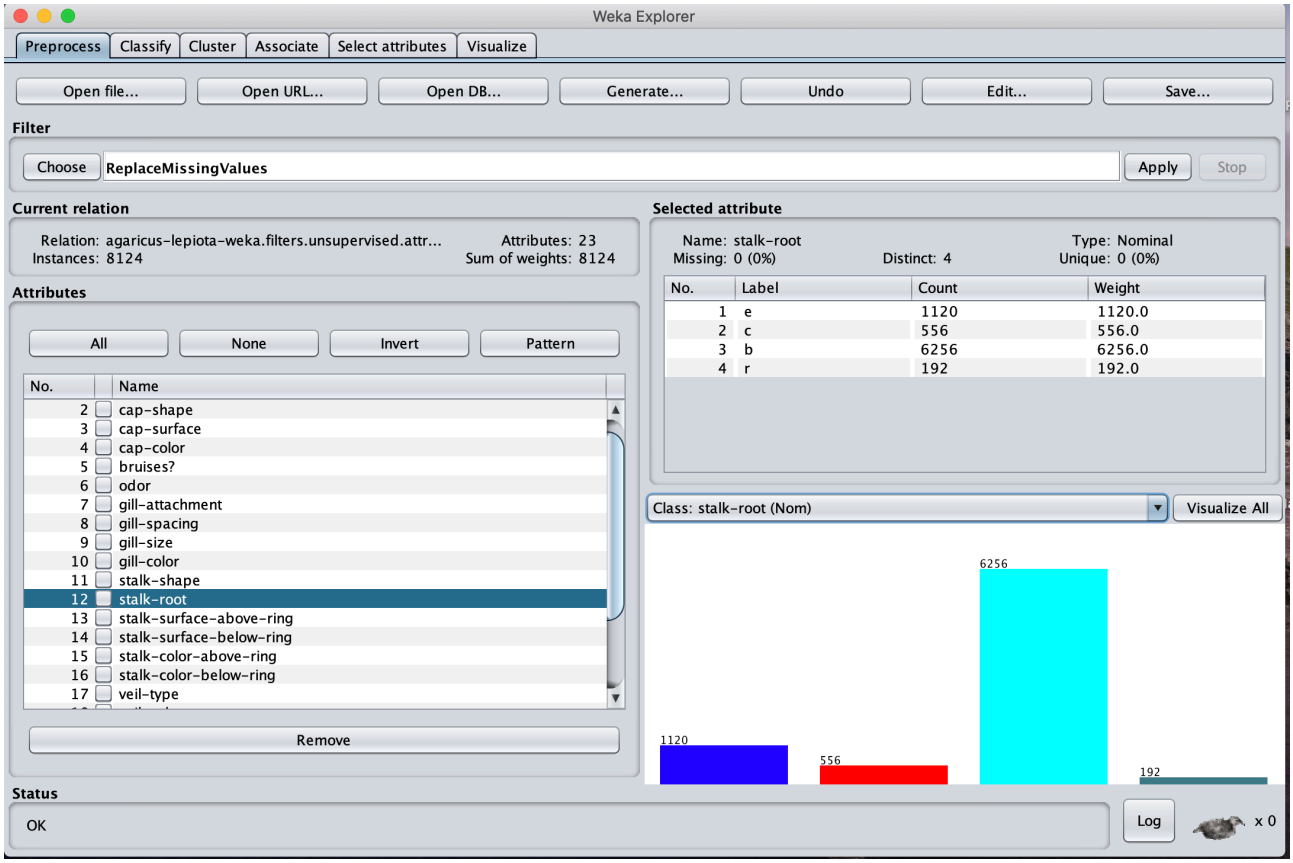
Data preparation

1. I fixed missing attribute by using "replacemissingvalue" function.

Before:



After:



2. I used attribute ranking to check which attribute is not necessary to use.

```
Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 1 Labels):
        Correlation Ranking Filter
Ranked attributes:
 0.5792    6 odor
 0.54      9 gill-size
 0.5015    5 bruises?
 0.4928   13 stalk-surface-above-ring
 0.4341   14 stalk-surface-below-ring
 0.4131   20 ring-type
 0.3985   21 spore-print-color
 0.3484    8 gill-spacing
 0.3172   12 stalk-root
 0.2945   22 population
 0.242    10 gill-color
 0.2227   15 stalk-color-above-ring
 0.2187   16 stalk-color-below-ring
 0.1833   19 ring-number
 0.1675   23 habitat
 0.1396   18 veil-color
 0.1292    7 gill-attachment
 0.1213    3 cap-surface
 0.102    11 stalk-shape
 0.0753    4 cap-color
 0.0464    2 cap-shape
 0        17 veil-type

Selected attributes: 6,9,5,13,14,20,21,8,12,22,10,15,16,19,23,18,7,3,11,4,2,17 : 22
```

Then I delete attribute 17.

Modelling

These four technologies have been improved to varying degrees. The data obtained are more readable and comprehensible. The pipeline model simulates the process of the optimized data set processed by four algorithms in cross validation. We use classAssigner to identify age as a class and use example to balance the dataset.

Evaluation

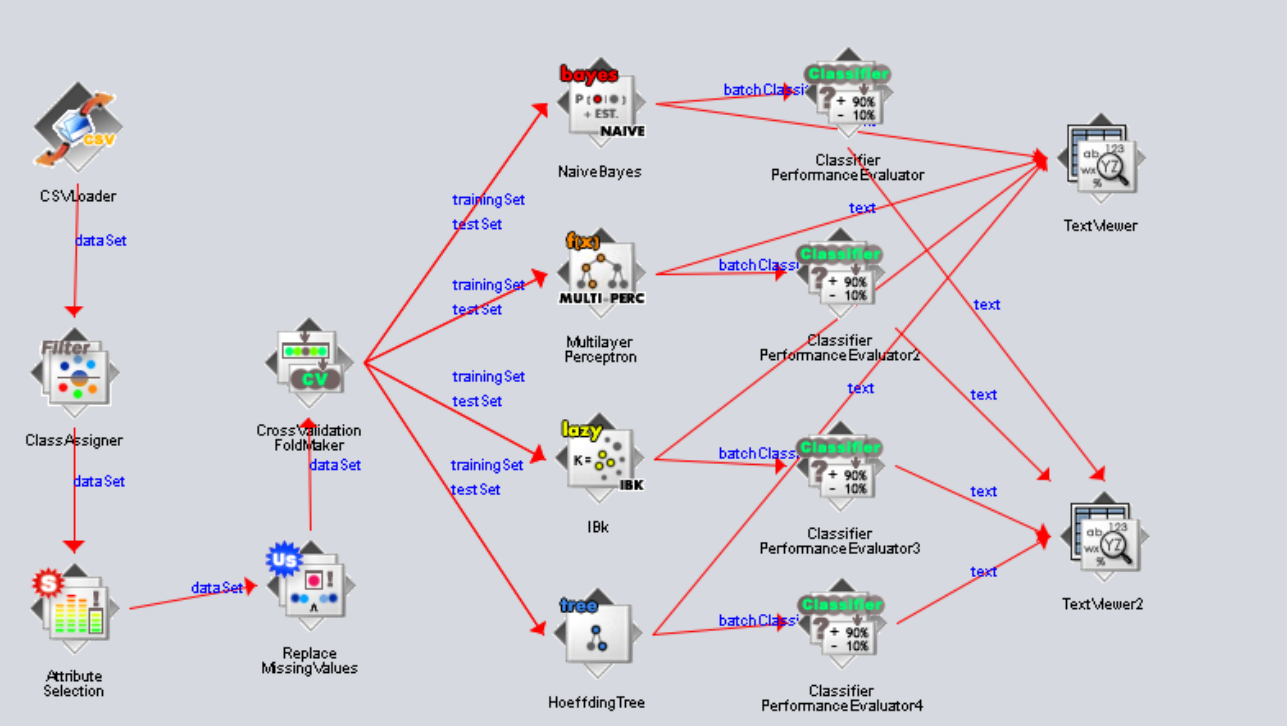
Yes.I used four methods :Hoeffding Tree, Multilayer Perceptron , NaiveBayes , IBk .

Deployment

I already fixed missing value and delete unnecessary attribute, so no more effort needed now.

2.3 Completion 2: Use the pipeline to reevaluate the selected techniques in Part 2.1 used to classify the dataset [20 marks]

My pipeline:



Hoeffding Tree:

```
=== Evaluation result ===

Scheme: HoeffdingTree
Options: -L 2 -S 1 -E 1.0E-7 -H 0.05 -M 0.01 -G 200.0 -N 0.0
Relation: agaricus-lepiota-weka.filters.unsupervised.attribute.ClassAssigner-Clast-weka.filters.supervised.attribu

=== Summary ===

Correctly Classified Instances        5129               63.1339 %
Incorrectly Classified Instances      2995               36.8661 %
Kappa statistic                          0.5385
Mean absolute error                      0.1063
Root mean squared error                  0.2867
Relative absolute error                 49.9074 %
Root relative squared error             87.8431 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.785    0.033    0.529      0.785    0.632      0.625    0.980     0.555     u
                 0.541    0.058    0.770      0.541    0.635      0.548    0.890     0.794     g
                 0.911    0.036    0.484      0.911    0.632      0.648    0.980     0.484     m
                 0.630    0.011    0.974      0.630    0.765      0.696    0.911     0.892     d
                 0.486    0.159    0.334      0.486    0.396      0.282    0.833     0.328     p
                 1.000    0.004    0.846      1.000    0.916      0.918    1.000     1.000     w
                 0.820    0.125    0.428      0.820    0.563      0.531    0.936     0.622     l
Weighted Avg.    0.631    0.058    0.733      0.631    0.651      0.582    0.905     0.732

=== Confusion Matrix ===

   a    b    c    d    e    f    g   <-- classified as
 289   63    0   16    0    0    0 |   a = u
 255 1161  276   19  429    8    0 |   b = g
   0   18  266    0    0    8    0 |   c = m
   2  147    0 1983  555    0  461 |   d = d
   0  108    8    4  556   19  449 |   e = p
   0    0    0    0    0  192    0 |   f = w
   0   10    0   14  126    0  682 |   g = l
```

IBk:

```
=== Evaluation result ===

Scheme: IBk
Options: -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\
Relation: agaricus-lepiota-weka.filters.unsupervised.attribute.ClassAssigner-Clast-weka.filters.supervised.attr

=== Summary ===

Correctly Classified Instances        4101               50.4801 %
Incorrectly Classified Instances      4023               49.5199 %
Kappa statistic                          0.3228
Mean absolute error                      0.1066
Root mean squared error                  0.2475
Relative absolute error                 50.0124 %
Root relative squared error             75.8338 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.446    0.034    0.383      0.446    0.412      0.383    0.976     0.624     u
                 0.550    0.222    0.472      0.550    0.508      0.314    0.893     0.788     g
                 0.031    0.026    0.042      0.031    0.035      0.005    0.965     0.343     m
                 0.709    0.235    0.656      0.709    0.681      0.468    0.881     0.868     d
                 0.052    0.103    0.076      0.052    0.061      -0.061   0.724     0.233     p
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     w
                 0.317    0.046    0.439      0.317    0.368      0.314    0.910     0.561     l
```

Multilayer Perceptron:

```
Scheme: MultilayerPerceptron
Options: -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation: agaricus-lepiota-weka.filters.unsupervised.attribute.ClassAssigner-Clast-weka.filters.supervisec

=== Summary ===

Correctly Classified Instances        5183               63.7986 %
Incorrectly Classified Instances      2941               36.2014 %
Kappa statistic                          0.5261
Mean absolute error                      0.0981
Root mean squared error                  0.2271
Relative absolute error                 46.0215 %
Root relative squared error             69.5796 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.598    0.024    0.546      0.598    0.571      0.550    0.983     0.695     u
                 0.632    0.097    0.701      0.632    0.665      0.554    0.931     0.845     g
                 0.421    0.021    0.423      0.421    0.422      0.400    0.976     0.404     m
                 0.732    0.089    0.838      0.732    0.782      0.662    0.927     0.910     d
                 0.338    0.119    0.317      0.338    0.327      0.213    0.837     0.356     p
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     w
                 0.720    0.101    0.449      0.720    0.553      0.507    0.940     0.653     l
Weighted Avg.    0.638    0.089    0.665      0.638    0.646      0.548    0.923     0.763

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
  220  148    0    0    0    0    0 |    a = u
  183 1357  167  145  296    0    0 |    b = g
    0  168  123    0    1    0    0 |    c = m
    0   94    0 2305  372    0  377 |    d = d
    0  168    1  231  387    0  357 |    e = p
    0    0    0    0    0  192    0 |    f = w
    0    0    0   68  165    0  599 |    g = l
```

NaiveBayes:

```
Scheme: NaiveBayes
Relation: agaricus-lepiota-weka.filters.unsupervised.attribute.ClassAssigner-Clast-weka.filters.superv

=== Summary ===

Correctly Classified Instances        5129               63.1339 %
Incorrectly Classified Instances      2995               36.8661 %
Kappa statistic                          0.5385
Mean absolute error                      0.1063
Root mean squared error                  0.2867
Relative absolute error                 49.9074 %
Root relative squared error             87.8431 %
Total Number of Instances             8124

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.785    0.033    0.529      0.785    0.632      0.625    0.980     0.555     u
                 0.541    0.058    0.770      0.541    0.635      0.548    0.890     0.794     g
                 0.911    0.036    0.484      0.911    0.632      0.648    0.980     0.484     m
                 0.630    0.011    0.974      0.630    0.765      0.696    0.911     0.892     d
                 0.486    0.159    0.334      0.486    0.396      0.282    0.833     0.328     p
                 1.000    0.004    0.846      1.000    0.916      0.918    1.000     1.000     w
                 0.820    0.125    0.428      0.820    0.563      0.531    0.936     0.622     l
Weighted Avg.    0.631    0.058    0.733      0.631    0.651      0.582    0.905     0.732

=== Confusion Matrix ===

    a    b    c    d    e    f    g   <-- classified as
  289   63    0   16    0    0    0 |    a = u
  255 1161  276   19  429    8    0 |    b = g
    0   18  266    0    0    8    0 |    c = m
    2  147    0 1983  555    0  461 |    d = d
    0  108    8    4  556   19  449 |    e = p
    0    0    0    0    0  192    0 |    f = w
    0   10    0   14  126    0  682 |    g = l
```

Techniques / results (before):

	Naive Bayes	Multilayer Perceptron	Hoeffding Tree	IBk
Correct	95.8272	99.0153	81.8808	99.9138
Incorrect	4.1728	0.9847	18.1192	0.0862

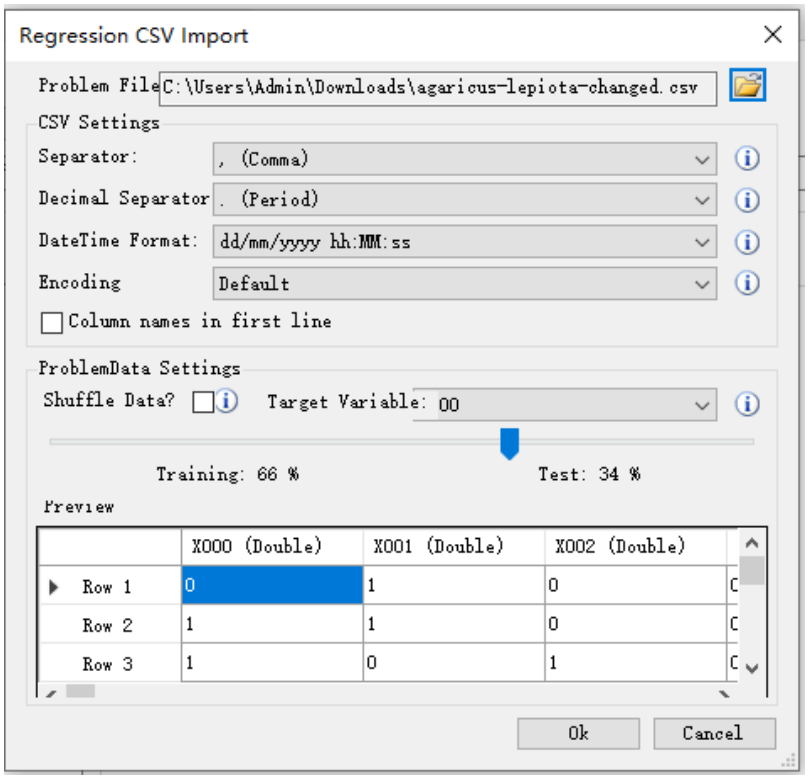
Techniques / results (after):

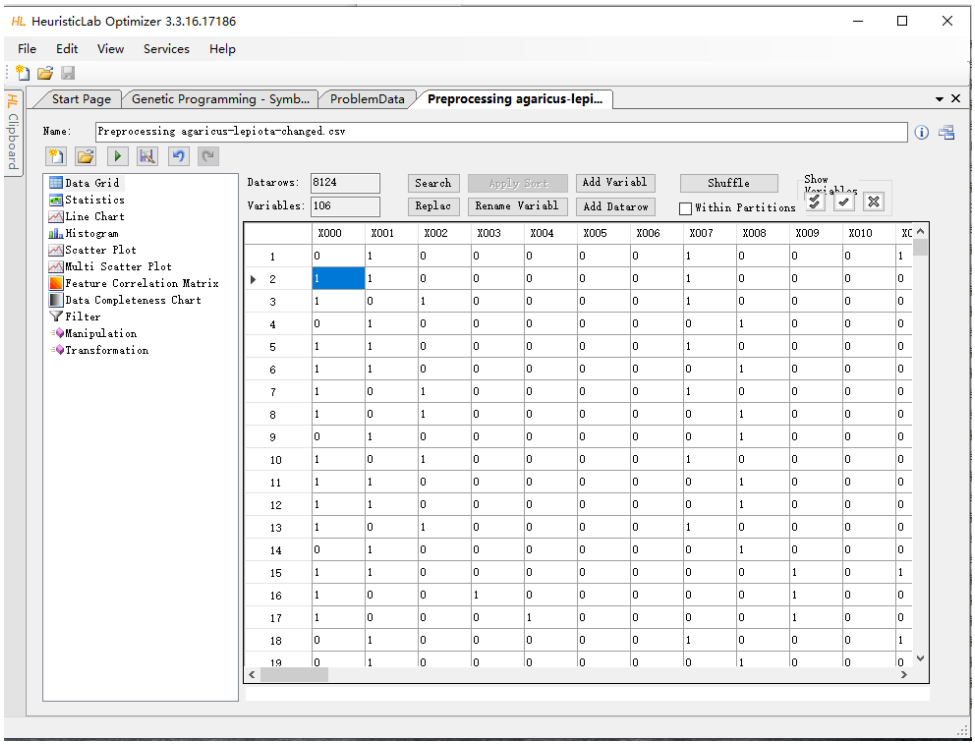
	Naive Bayes	Multilayer Perceptron	Hoeffding Tree	IBk
Correct	63.1339	63.7986	63.1339	50.4801
Incorrect	36.8661	36.2014	36.8661	49.5199

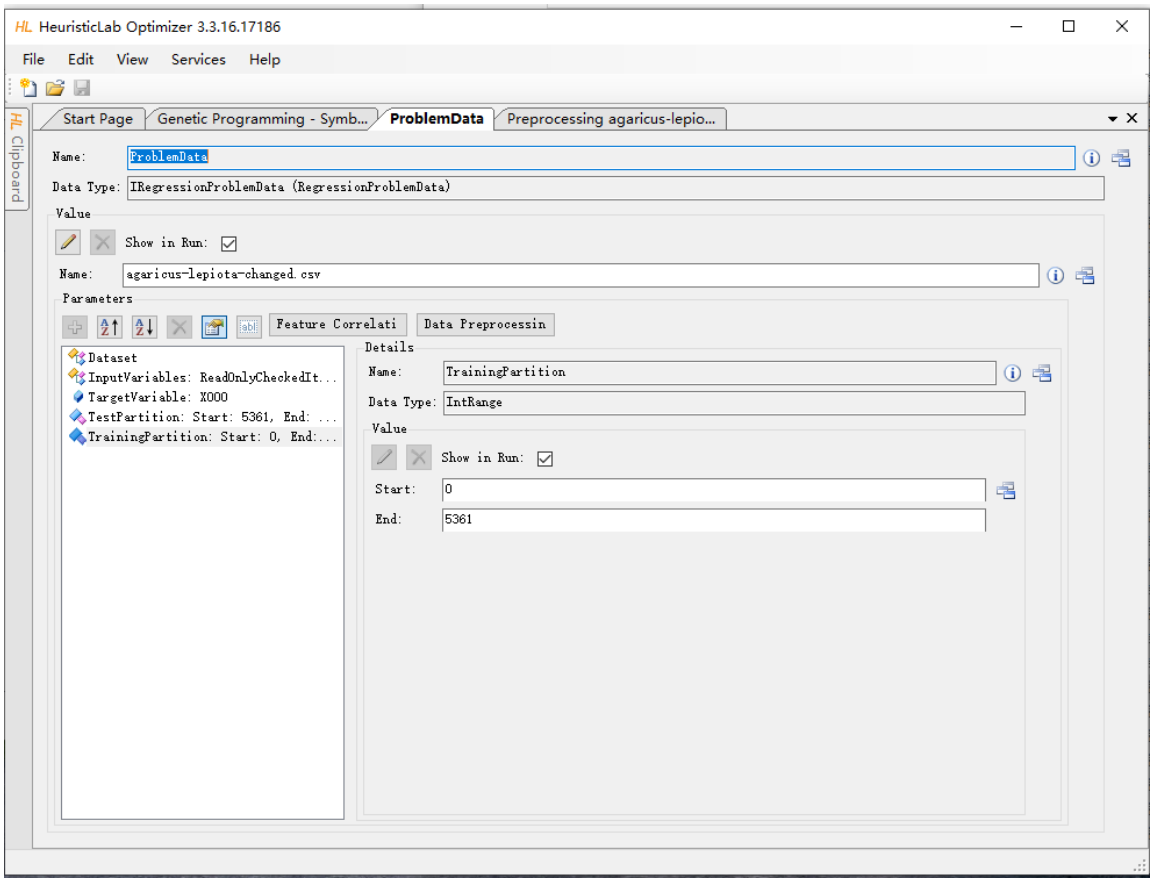
The accuracy decrease because cross-validation is more reasonable way to check the correctness than only using validation(using training set).

The reason why cross validation is used is that it can divide the original data into several files, one of which is used as the test data set and the other as the training set, which can check the accuracy of the data set more accurately and reasonably. It's better than just using the training set.

2.4 Challenge: Use the HeuristicLab to evaluate the Evolutionary Computation tribe on the dataset in Part 2.1 to classify the dataset [15 marks]




HeuristicLab Optimizer 3.3.16.17186
File Edit View Services Help
Start Page
Genetic Programming - Symb...
ProblemData
Preprocessing agaricus-lepi...
Name: Preprocessing agaricus-lepiota-changed.csv
Data Grid
Statistics
Line Chart
Histogram
Scatter Plot
Multi Scatter Plot
Feature Correlation Matrix
Data Completeness Chart
Filter
Manipulation
Transformation
Datarows: 8124
Variables: 106
Search
Replac
Apply Sort
Rename Variabl
Add Variabl
Add Datarow
Shuffle
Within Partitions
X000 X001 X002 X003 X004 X005 X006 X007 X008 X009 X010


HeuristicLab Optimizer 3.3.16.17186
File Edit View Services Help
Start Page
Genetic Programming - Symb...
ProblemData
Preprocessing agaricus-lepio...
Name: ProblemData
Data Type: IRegressionProblemData (RegressionProblemData)
Value
Show in Run:
Name: agaricus-lepiota-changed.csv
Parameters
Feature Correlati
Data Preprocessin
Dataset
InputVariables: ReadOnlyCheckedIt...
TargetVariable: X000
TestPartition: Start: 5361, End: ...
TrainingPartition: Start: 0, End:...
Details
Name: TrainingPartition
Data Type: IntRange
Value
Show in Run:
Start: 0
End: 5361