Artificial Intelligence and Machine Learning Project: Week6 Group Assignment

Group 9

Student ID:
2226442
2152050
1943618
2143752
2226250
2090739

Date: 08/03/2021

Dataset description and reading in Weka (T1)
The size of this dataset is 299. The attributes are: Age, Anaemia, Creatinine Phosphokinase, Diabetes, Ejection Fraction, High Blood Pressure, Platelets, Serum Creatinine, Serum Sodium, Sex, Smoking, Time. Among these attributes, Sex is a binary attribute. Anaemia, Diabetes, High Blood Pressure and Smoking are boolean attributes. The rest of the attributes are numeric attributes.
The range of age is 40 to 95, and the majority of people are located in range 64 to 70. In the case of anaemia, there are slightly more people who don't have anaemia than those who have. As the amount of Creatinine Phosphokinase increases, the number of people decreases, and the range is 23 to 7861. There are more people who don’t have diabetes than who have diabetes. For ejection fraction, range is 14 to 80 and majority of people are located in the range 32 to 38. There are more people who don’t have high blood pressure than who don’t have. The range of platelets is 25100 to 850000 and the majority of people are located in the range 231325 to 282881.25. In the case of serum creatinine, it’s range is 0.5 to 9.4, and when serum creatinine increases, the number of people decreases. For serum sodium, the range of it is 113 to 148, and the majority of people are located in the range 138 to 141. There are more male than female, and there are more people who are not smokers than who are smokers. The range of the time is 4 to 285 and the majority of people are located in range 84 to 124.

The learning task (T2)
We consider the attributes a patient has which are most associated with Death Event as inputs. And we consider the Death Event as output.

Data preprocessing (T3)
The preprocessing we performed on the dataset is as follows:

Apply the “NumericToNominal” filter to all the binary or boolean input attributes such as Sex, Anaemia, Diabetes, High Blood Pressure and Smoking and also the output which is the Death Event. In this way, WEKA can recognize them as labels rather than continuous values. Remove the attributes: Smoking, Sex, Diabetes and Time from the dataset. This is because by using CorrelationAttributeEveluater in WEKA, it is found that smoking, sex and diabetes have the lowest correlations to DEATH_EVENT, which are all close to or lower than 0.01. Thus removing them would increase the efficiency and accuracy of our prediction model.

Although the attribute Time relates most closely to DEATH_EVENT, it is also removed since this misleading attribute may result in the overfitting of the learning models. DEATH_EVENT means whether the person deceased during the follow-up period or not, which indicates that the time is depending on DEATH_EVENT(i.e., if a patient gets heart failure, then the follow-up time will be shorter), not reversely. Hence time can not be used to predict the DEATH_EVENT.

Experimental Design (T4)
(first point): List of supervised learning algorithms:
1. IBK(K-nearest neighbor)
2. J48(decision tree)
3. Logistic (logistic regression)
4. Naive Bayes
5. SGD (stochastic gradient descent)
6. MultilayerPerceptron (neural network)

(second point): The model evaluation technique we use is 10-folds-Cross-validation. By using this technique, the data will split into 90% training and 10% testing. After the experiment is done, we then choose another 90% of data as train data and the rest as test data and do the experiment again. This will repeat for 10 times. By doing so, every part of the data has been used as test data. It helps us better use our data, and it gives us much more information about our algorithm performance. It is better than a percentage split because the percentage split will only split the data into testing and training once.

The strategy we use for tuning hyper-parameters is manual search. On the one hand, this strategy is not time consuming; On the other hand, by using this strategy, we can learn the behavior of hyperparameters by heart and use your knowledge in another project which can be helpful for us in the future.

T5 (Appendix)

Result comparison and discussion (T6)
Result comparison and discussion:
By comparing the model's performance with this dataset, we can see that:

IBK < Naive Bayes < MultilayerPerceptron < J48 < Logistic Regression

The reasons that lead to this result are listed below.

(comparison between IBK and Logistic Regression)
KNN is much slower than logistic regression, so logistic regression is more efficient than IBK. Also, logistic regression needs less memory than IBK. Furthermore, IBK is not a parametric model, but logistic regression is a parametric model, so in this assignment logistic regression is a better algorithm than IBK.

(comparison between SGD and Logistic Regression)
Both model 5 and the Logistic model used logistic regression as the loss function. Hence it might be the method of updating rate that resulted in the slightly better performance of the Logistic model. When using stochastic gradient descent, only a batch of sample points was used to update the weight in each iteration instead of using all the data as in normal gradient descent. Since the size of the dataset is relatively small (299), it may increase the inaccuracy by using SGD although this method may be faster.

(comparison between Naive Bayes and Logistic Regression)
Logistic regression is usually used when the output is a boolean, which in this problem the output is in boolean. Naive Bayes assumes that all attributes are independent to each other but actually there is no absolute that all attributes are independent to each other. Therefore, when Naive Bayes assumes that all attributes are independent to each other when some of them are not, this could lead to higher bias than Logistic regression. From this, Naive Bayes would have less accuracy than Logistic regression.

(comparison between DT and Logistic Regression)
Logistic regression is better than decision tree in analyzing the whole data structure, and decision tree is better than logistic regression in analyzing the local data structure. Also, Logistic regression is good at analyzing linear relationships, but the decision tree has a poor grasp of linear relationship.

(comparison between Multilayer perceptron and Logistic Regression)
Neural networks depend a lot on training data. This leads to the problem of overfitting and generalization. The mode relies more on the training data and may be tuned to the data. Neural networks are good to model with a large number of inputs which is not similar to our data set.

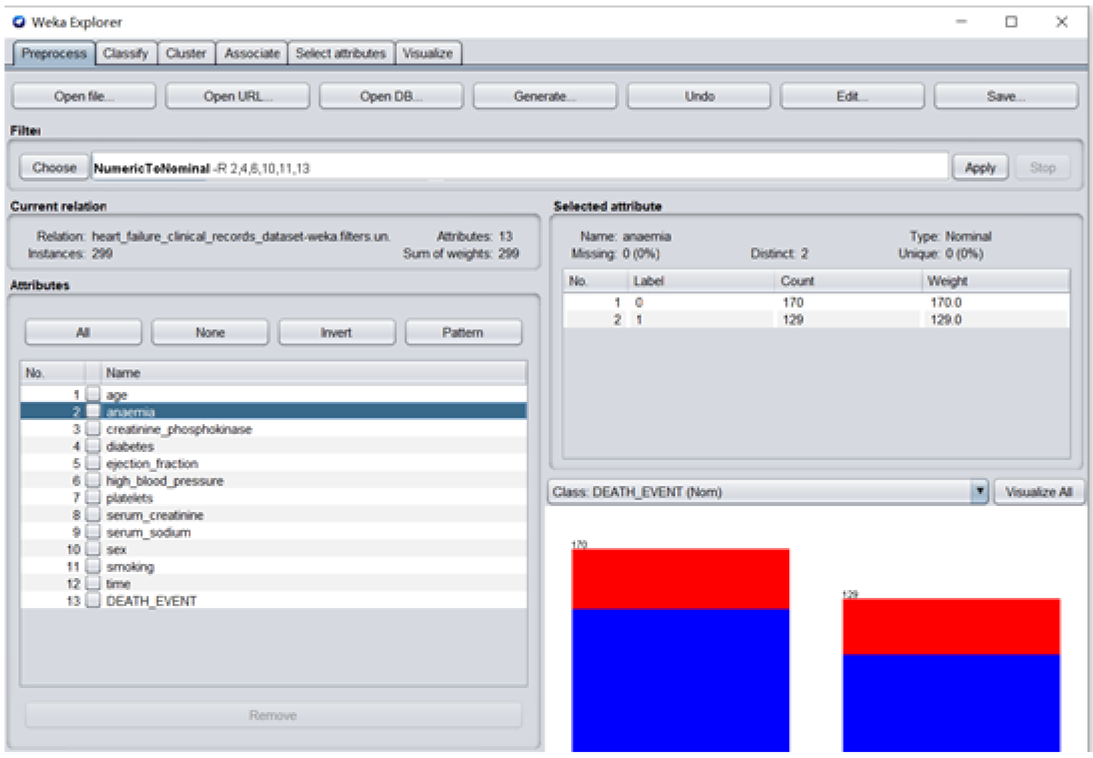
Conclusion and recommendations (T7)
We observed data and processed it,and formulated learning tasks and prepared to establish the model. Also, in T3 we removed low relevant inputs, and got different 6 algorithms. Finally the best models are obtained for each model, and by comparing the six models, we concluded that logistic regression has the best F-measures. Therefore, we recommend using Logistic to use machine learning to predict heart disease.

Appendix

T3

(1)

Attributes before preprocess



Filter

NumericToNominal -R 2,4,6,10,11,13

Current relation

Relation: heart_failure_clinical_records_dataset-weka.filters.un... Attributes: 13
Instances: 299 Sum of weights: 299

Attributes

No.	Name
1	age
2	anaemia
3	creatinine_phosphokinase
4	diabetes
5	ejection_fraction
6	high_blood_pressure
7	platelets
8	serum_creatinine
9	serum_sodium
10	sex
11	smoking
12	time
13	DEATH_EVENT

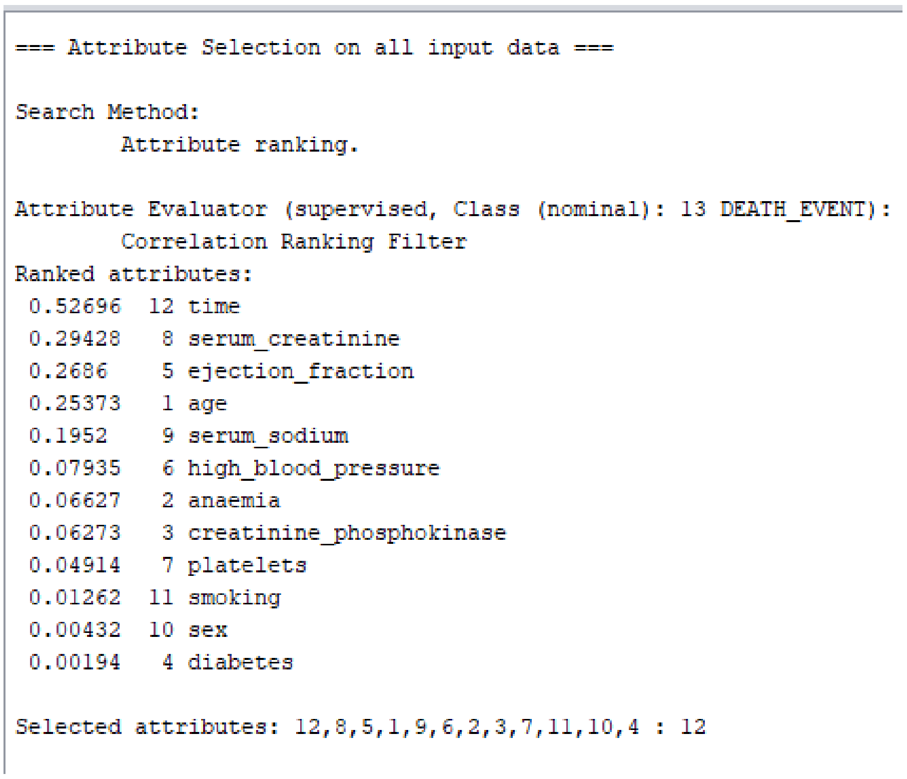
Selected attribute

Name: anaemia Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	170	170.0
2	1	129	129.0

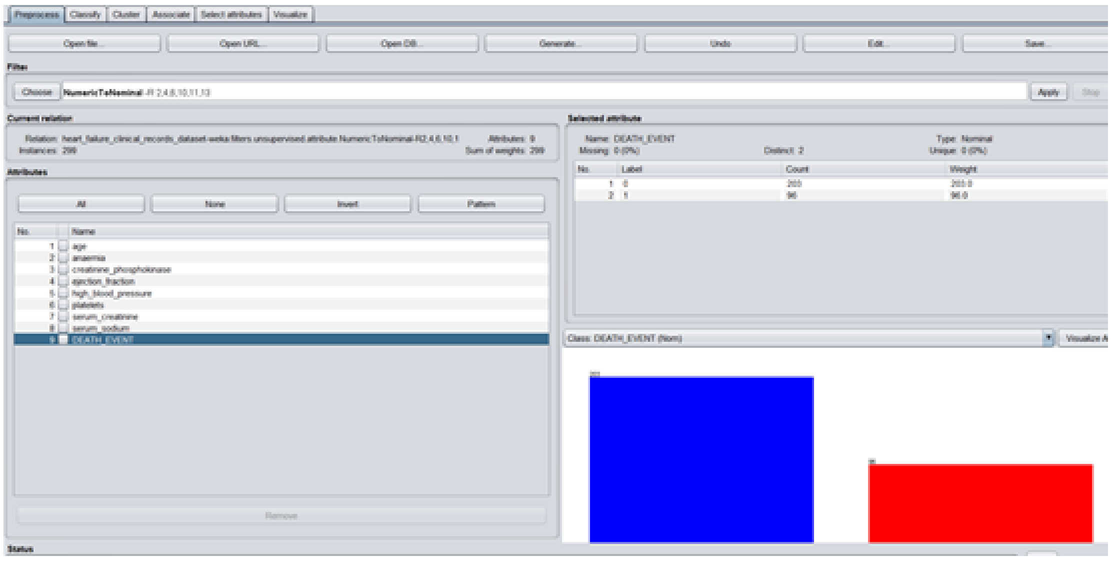
Class: DEATH_EVENT (Nom)

(2)



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 13 DEATH_EVENT):
        Correlation Ranking Filter
Ranked attributes:
 0.52696   12 time
 0.29428    8 serum_creatinine
 0.2686     5 ejection_fraction
 0.25373    1 age
 0.1952     9 serum_sodium
 0.07935    6 high_blood_pressure
 0.06627    2 anaemia
 0.06273    3 creatinine_phosphokinase
 0.04914    7 platelets
 0.01262   11 smoking
 0.00432   10 sex
 0.00194    4 diabetes

Selected attributes: 12,8,5,1,9,6,2,3,7,11,10,4 : 12
```

Attributes after preprocess



Filter: NumericToNominal -R 2,4,8,10,11,13

Current relation
Relation: heart_failure_clinical_records_dataset-weka.filters.unsupervised.attribute.NumericToNominal-R2,4,8,10,1 Attributes: 9
Instances: 299 Sum of weights: 299

Attributes

No.	Name
1	age
2	anaemia
3	creatinine_phosphokinase
4	ejection_fraction
5	high_blood_pressure
6	platelets
7	serum_creatinine
8	serum_sodium
9	DEATH_EVENT

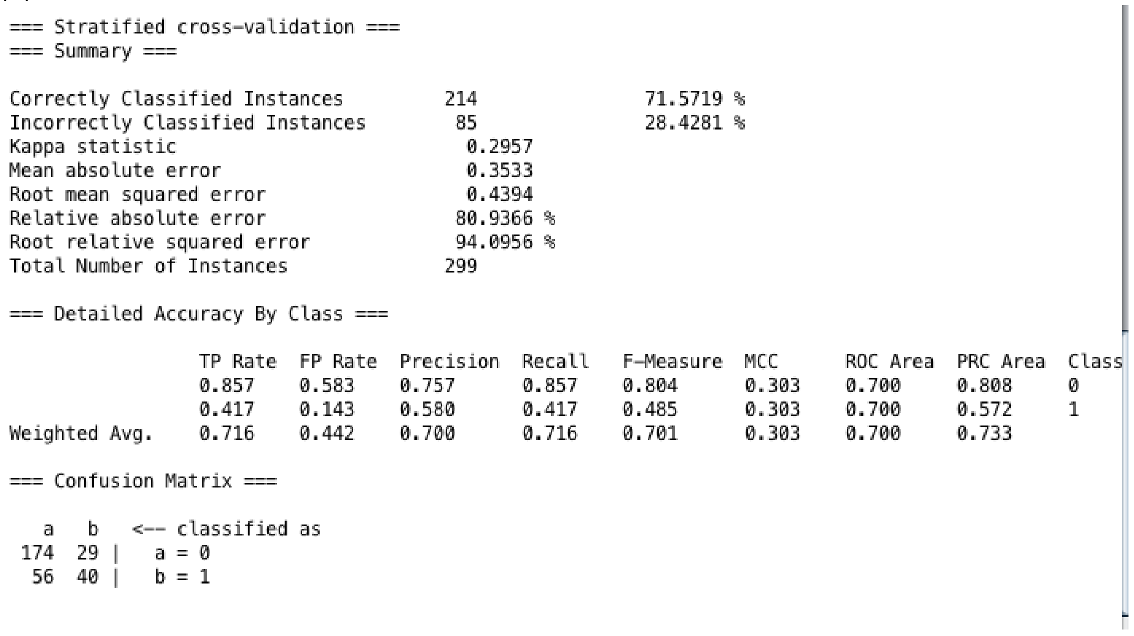
Selected attribute
Name: DEATH_EVENT Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

No.	Label	Count	Weight
1	0	203	203.0
2	1	96	96.0

Class: DEATH_EVENT (Nom)

T5

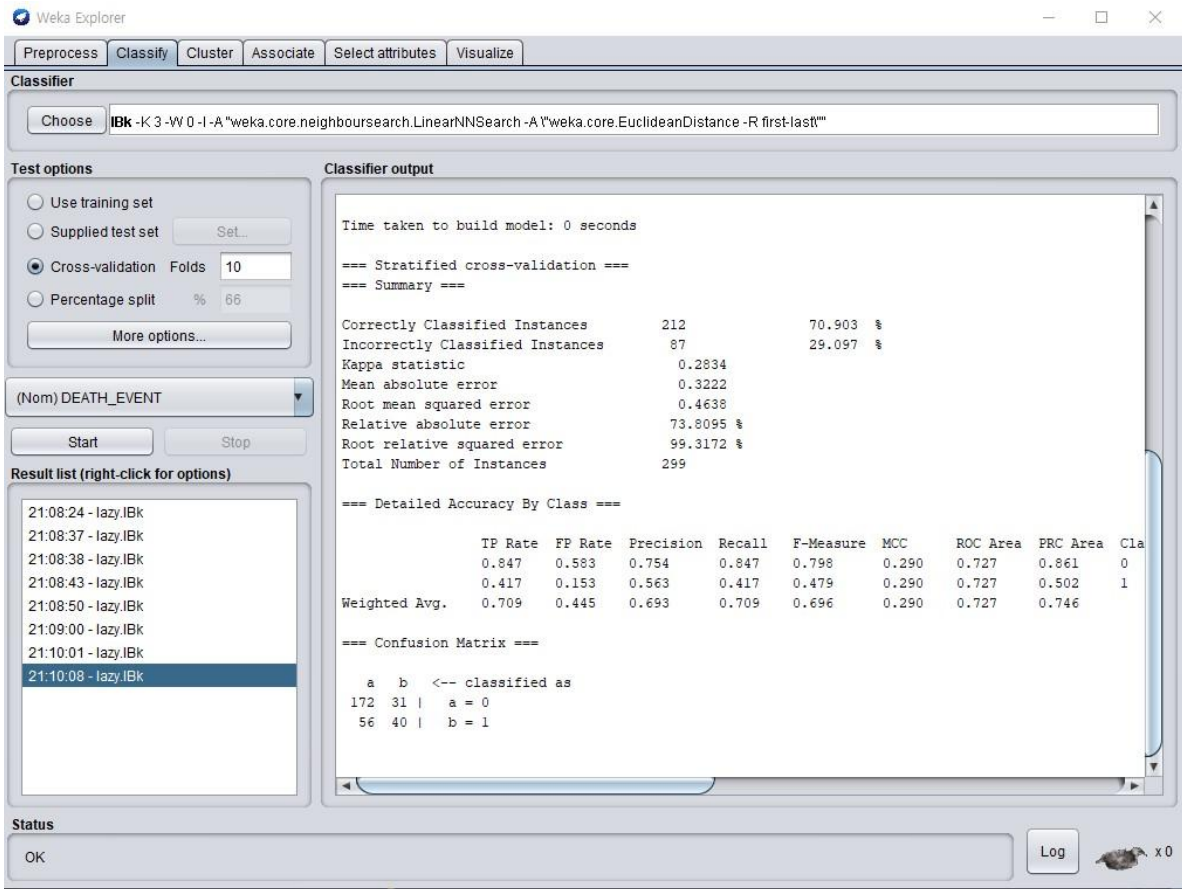
(1)



```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         214               71.5719 %
Incorrectly Classified Instances        85               28.4281 %
Kappa statistic                          0.2957
Mean absolute error                      0.3533
Root mean squared error                  0.4394
Relative absolute error                 80.9366 %
Root relative squared error             94.0956 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.857    0.583    0.757      0.857    0.804      0.303    0.700     0.808     0
                 0.417    0.143    0.580      0.417    0.485      0.303    0.700     0.572     1
Weighted Avg.    0.716    0.442    0.700      0.716    0.701      0.303    0.700     0.733

=== Confusion Matrix ===

   a   b   <-- classified as
 174  29 |   a = 0
  56  40 |   b = 1
```

Using Naive Bayes: the hyper-parameter that were tuned are useSupervisedDiscretization and removing diabetes and sex from the attribute. For the value of useSupervisedDiscreization, its default value is False and tuning this by giving the value of useSupervisedDiscretization as True. This makes the F-measurement of this algorithm increased. Furthermore, by looking at the correlation ranking on the CorrelationAttributeEval, which evaluates how much the attribute is correlated between it's class and itself. From CorrelatuonAttributeEval shows that diabetes have the least correlation, following by sex. Therefore, deleting these attributes from the model, helps to increase the F-measurement or the accuracy of this prediction model.

(2)



IBk -K 3 -W 0 -I -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""

[✓ Cross-validation] Folds 10

Percentage split % 66

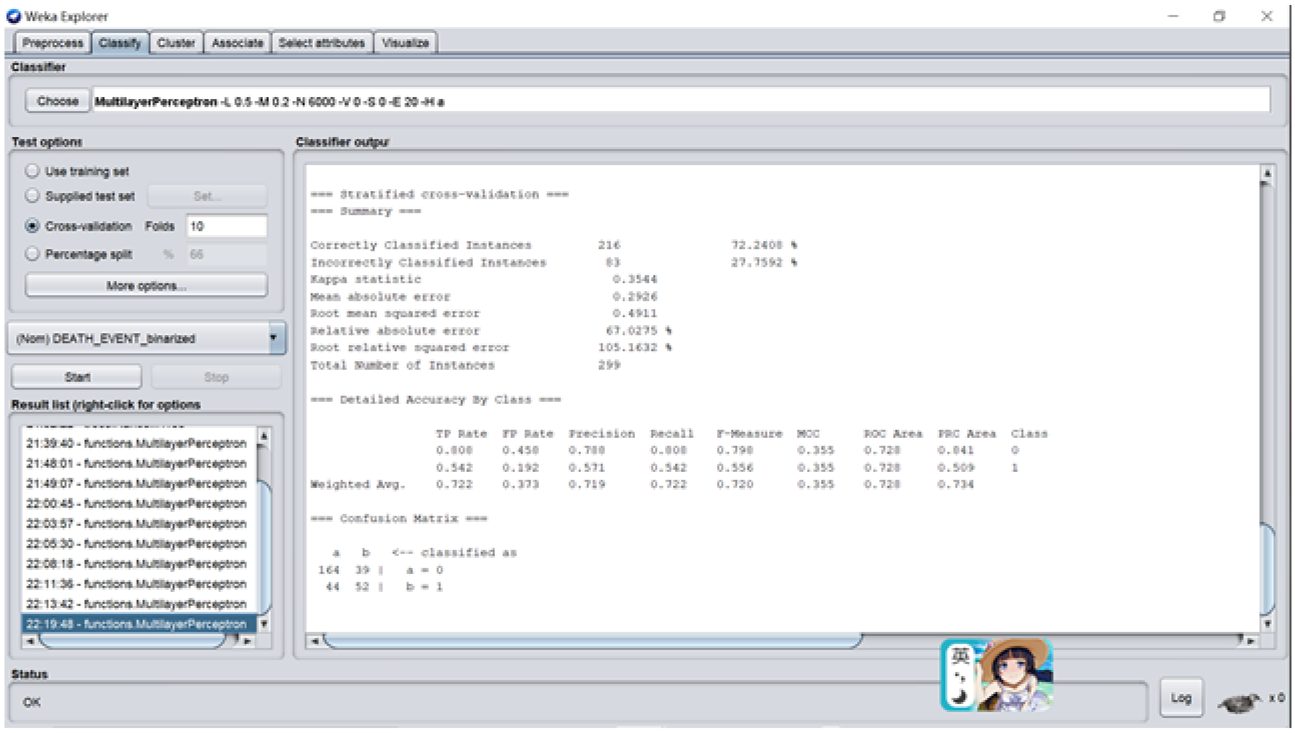
(Nom) DEATH_EVENT

```
21:08:24 - lazy.IBk
21:08:37 - lazy.IBk
21:08:38 - lazy.IBk
21:08:43 - lazy.IBk
21:08:50 - lazy.IBk
21:09:00 - lazy.IBk
21:10:01 - lazy.IBk
21:10:08 - lazy.IBk
```

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         212               70.903  %
Incorrectly Classified Instances        87               29.097  %
Kappa statistic                          0.2834
Mean absolute error                      0.3222
Root mean squared error                  0.4638
Relative absolute error                 73.8095 %
Root relative squared error             99.3172 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cla
                 0.847    0.583    0.754      0.847    0.798      0.290    0.727     0.861     0
                 0.417    0.153    0.563      0.417    0.479      0.290    0.727     0.502     1
Weighted Avg.    0.709    0.445    0.693      0.709    0.696      0.290    0.727     0.746

=== Confusion Matrix ===

   a   b   <-- classified as
 172  31 |   a = 0
  56  40 |   b = 1
```

Using IBK(K-nearest neighbor): As we explained in T3, it is better to delete 4 attributes, Time, Sex, Diabetes and Smoking. In addition, before starting tuning, we should normalize this value. And I found the best KNN is 3. If we choose less than 3 for KNN, overfitting can be shown. Also, if we choose bigger than 3 for KNN, underfitting can be shown. Furthermore, if we choose an even number for KNN two same numbers of different things could exist, so we should use an even number of KNN. Plus, weighting by distance makes the prediction more accurate. Finally with this tunning I got F-Measure value 0.696.

(3) (2090739)



MultilayerPerceptron -L 0.5 -M 0.2 -N 6000 -V 0 -S 0 -E 20 -H a

Use training set
Supplied test set
[✓ Cross-validation] Folds 10
Percentage split % 66

(Nom) DEATH_EVENT_binarized

Result list (right-click for options)

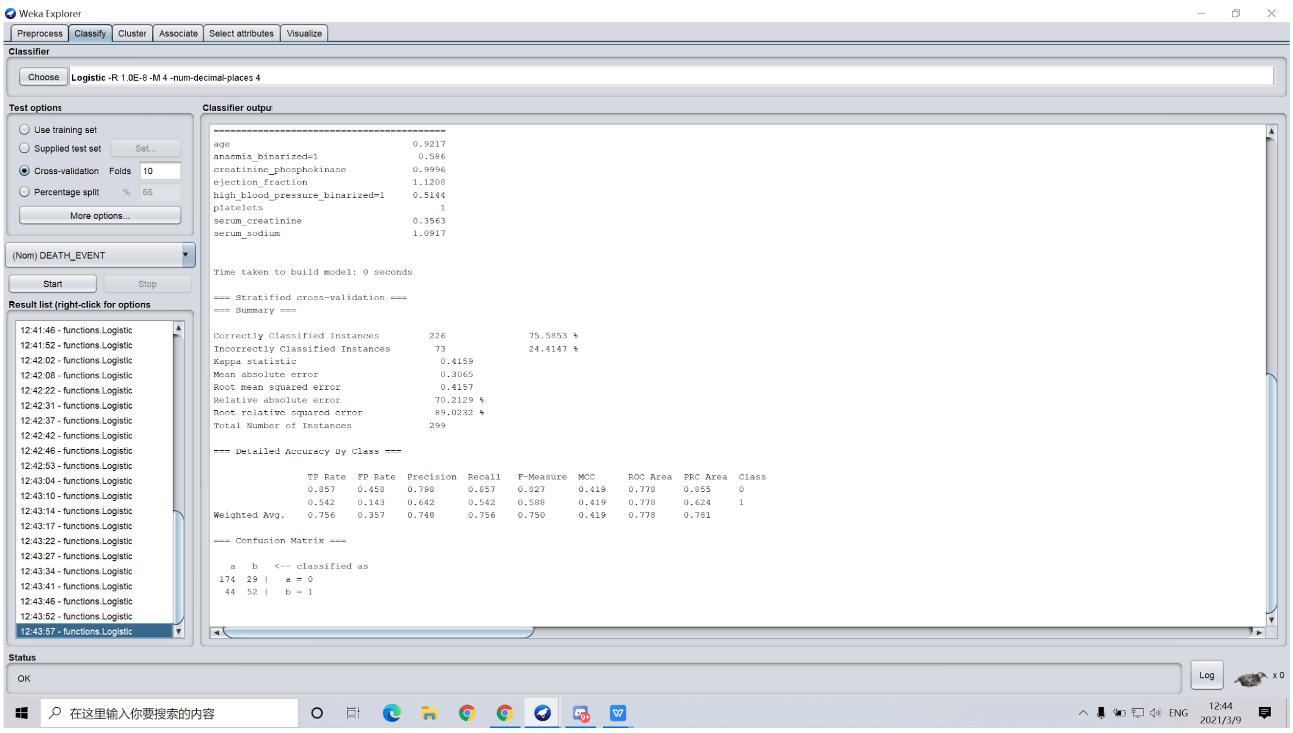
```
21:39:40 - functions.MultilayerPerceptron
21:48:01 - functions.MultilayerPerceptron
21:49:07 - functions.MultilayerPerceptron
22:00:45 - functions.MultilayerPerceptron
22:03:57 - functions.MultilayerPerceptron
22:05:30 - functions.MultilayerPerceptron
22:08:18 - functions.MultilayerPerceptron
22:11:36 - functions.MultilayerPerceptron
22:13:42 - functions.MultilayerPerceptron
22:19:48 - functions.MultilayerPerceptron
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         216               72.2408 %
Incorrectly Classified Instances        83               27.7592 %
Kappa statistic                          0.3544
Mean absolute error                      0.2926
Root mean squared error                  0.4911
Relative absolute error                 67.0275 %
Root relative squared error            105.1632 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.808    0.458    0.788      0.808    0.798      0.355    0.728     0.841     0
                 0.542    0.192    0.571      0.542    0.556      0.355    0.728     0.509     1
Weighted Avg.    0.722    0.373    0.719      0.722    0.720      0.355    0.728     0.734

=== Confusion Matrix ===

   a   b   <-- classified as
 164  39 |   a = 0
  44  52 |   b = 1
```

Status
OK

Using Multilayer perceptron :In this model, these attributes are deleted: Diabetes, sex, smoking and time, keep two input values: and make to nominal, here are two input variables: anaemia and ejection. It has been found that when the training time has changed, the learning rate has been kept, the total accuracy will be kept. Then if the learning rate reduces, the total accuracy will reduce (when keeping the training data in 3000) However, when the training time is changed to 5000, total accuracy has been reduced. Also, when the training data is greater or equal to 5000 and learning rate is greater or equal to 0.5, the total accuracy becomes higher. If the training time grows to 10000, the process will take more time. So, the training time has been kept to 6000 and 0.5 rates of learning.

(4) (2226250)


Classifier: Logistic -R 1.0E-8 -M 4 -num-decimal-places 4

Test options: Cross-validation Folds 10

(Nom) DEATH_EVENT

Result list (right-click for options)

```
12:41:46 - functions.Logistic
12:41:52 - functions.Logistic
12:42:02 - functions.Logistic
12:42:08 - functions.Logistic
12:42:22 - functions.Logistic
12:42:31 - functions.Logistic
12:42:37 - functions.Logistic
12:42:42 - functions.Logistic
12:42:46 - functions.Logistic
12:42:53 - functions.Logistic
12:43:04 - functions.Logistic
12:43:10 - functions.Logistic
12:43:14 - functions.Logistic
12:43:17 - functions.Logistic
12:43:22 - functions.Logistic
12:43:27 - functions.Logistic
12:43:34 - functions.Logistic
12:43:41 - functions.Logistic
12:43:46 - functions.Logistic
12:43:52 - functions.Logistic
12:43:57 - functions.Logistic
```

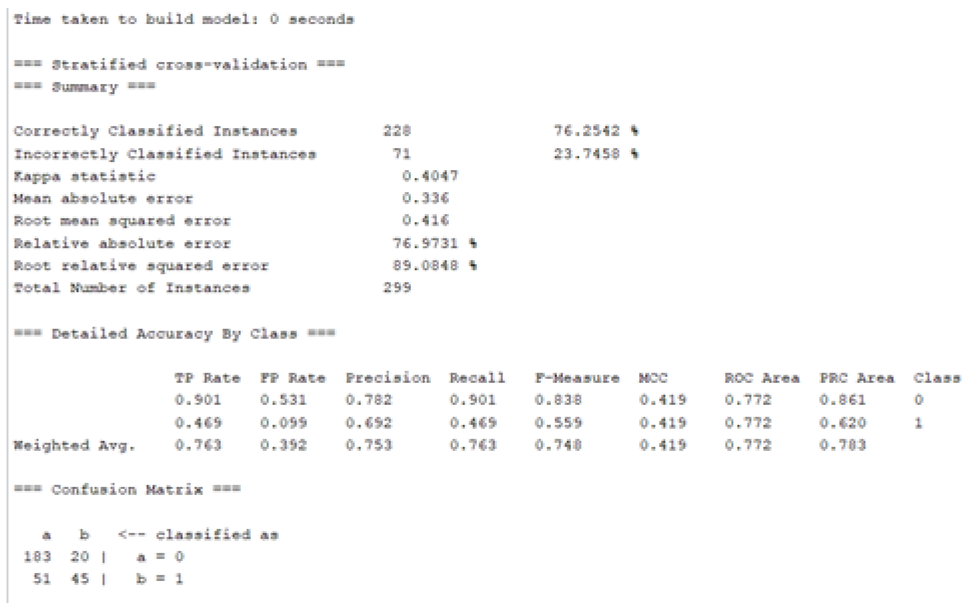
Classifier output

```
==========================================
age                                 0.9217
anaemia_binarized=1                  0.586
creatinine_phosphokinase            0.9996
ejection_fraction                   1.1208
high_blood_pressure_binarized=1     0.5144
platelets                                1
serum_creatinine                    0.3563
serum_sodium                        1.0917


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         226               75.5853 %
Incorrectly Classified Instances        73               24.4147 %
Kappa statistic                          0.4159
Mean absolute error                      0.3065
Root mean squared error                  0.4157
Relative absolute error                 70.2129 %
Root relative squared error             89.0232 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.857    0.458    0.798      0.857    0.827      0.419    0.778     0.855     0
                 0.542    0.143    0.642      0.542    0.588      0.419    0.778     0.624     1
Weighted Avg.    0.756    0.357    0.748      0.756    0.750      0.419    0.778     0.781

=== Confusion Matrix ===

   a   b   <-- classified as
 174  29 |   a = 0
  44  52 |   b = 1
```

Status: OK

Using Logistic Regression: In this model, the input attributes: Sex, Time, Diabetes and Smoking are removed. When tuning the hyper-parameters, there are four parameters I can choose. They are: batchsize, maxIts, numDecimalPlaces and ridge. The hyper-parameters I tuned is maxIts. The reason is that first of all, the batchsize and the numDecimalPlaces do not affect the performance of the algorithm. Secondly, the ridge will matter if the dataset is linear, which does not apply to the dataset we are using right now. Finally, the maxIts affect the performance of the algorithm because maxIts means the number of times of repeating the calculation. In my opinion, the larger it is, the more accurate the result is supposed to be. The default value of maxIts is -1, since I am using the strategy of manual search to tune the hyper-parameters, the values that are considered are from 1 to 10. In the end, the value that leads to the best performance is 4. I think this is because setting the times of iteration bigger than 4 will lead to the overfitting of the algorithm.

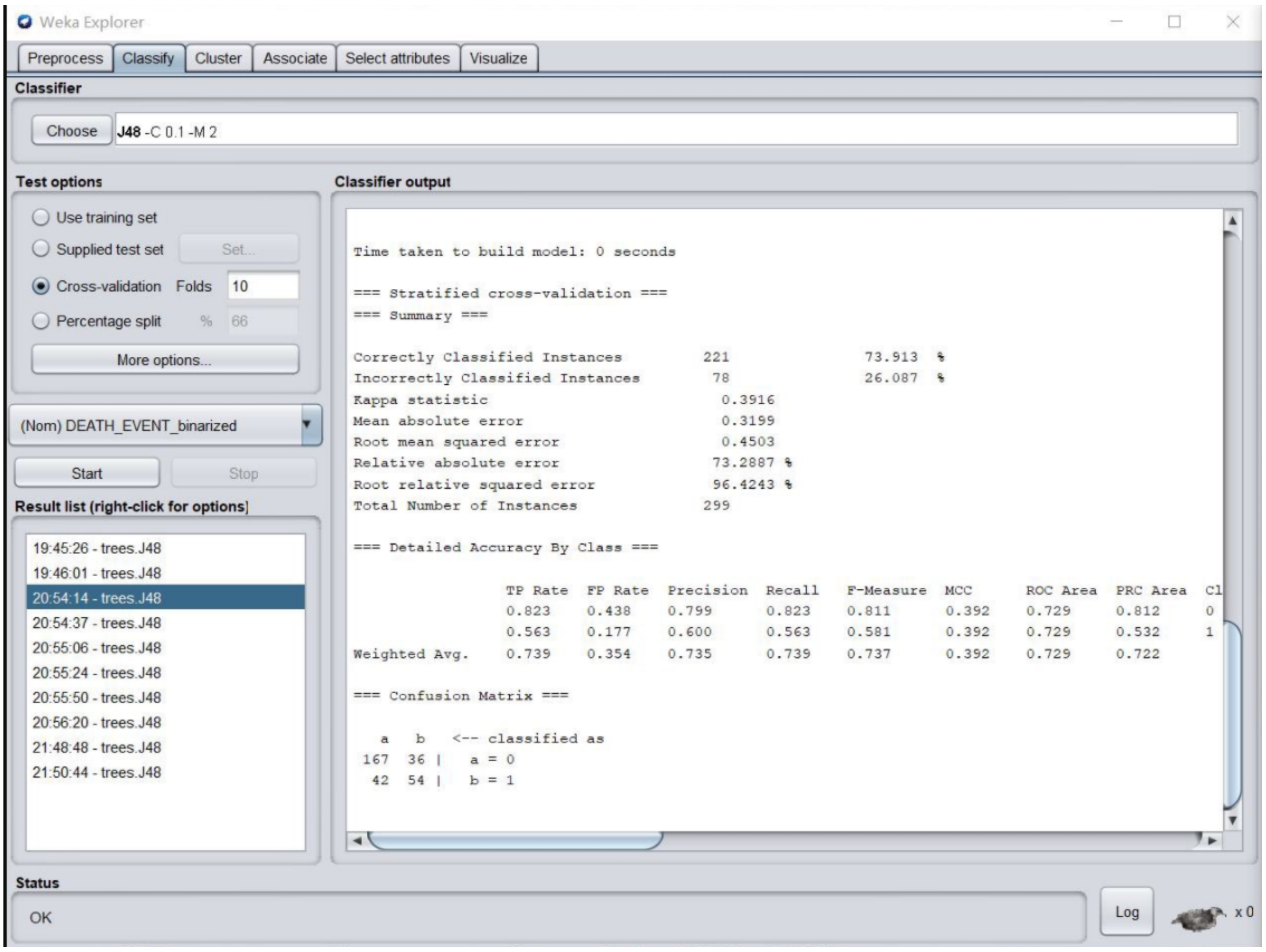
(5) (2152050)



```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         228               76.2542 %
Incorrectly Classified Instances        71               23.7458 %
Kappa statistic                          0.4047
Mean absolute error                      0.336
Root mean squared error                  0.416
Relative absolute error                 76.9731 %
Root relative squared error             89.0848 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.901    0.531    0.782      0.901    0.838      0.419    0.772     0.861     0
                 0.469    0.099    0.692      0.469    0.559      0.419    0.772     0.620     1
Weighted Avg.    0.763    0.392    0.753      0.763    0.748      0.419    0.772     0.783

=== Confusion Matrix ===

   a   b   <-- classified as
 183  20 |   a = 0
  51  45 |   b = 1
```

Using SGD:In this model, the logistic regression is used as the loss function to implement stochastic gradient descent. Using the default hyperparameter, the F-measure value is 0.724. When tuning the hyperparameter, batch size, epoch, lambda, the learning rate were taken into account. By doing experiments, it is discovered that reducing the epoch to 50, increasing the learning rate to 0.09 and remain other hyperparameter values unchanged could lead to the best model, with an F-measure value of 0.748.

(6) (2143752)



J48 -C 0.1 -M 2

Cross-validation Folds 10

Percentage split % 66

(Nom) DEATH_EVENT_binarized

```
Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         221               73.913  %
Incorrectly Classified Instances        78               26.087  %
Kappa statistic                          0.3916
Mean absolute error                      0.3199
Root mean squared error                  0.4503
Relative absolute error                 73.2887 %
Root relative squared error             96.4243 %
Total Number of Instances              299

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cl
                 0.823    0.438    0.799      0.823    0.811      0.392    0.729     0.812     0
                 0.563    0.177    0.600      0.563    0.581      0.392    0.729     0.532     1
Weighted Avg.    0.739    0.354    0.735      0.739    0.737      0.392    0.729     0.722

=== Confusion Matrix ===

   a   b   <-- classified as
 167  36 |   a = 0
  42  54 |   b = 1
```

Using J48: pruning（confidenceFactor，minNumObj ）is a confusing and complex topic, and it's not particularly enlightening. In fact, people don't recommend changing these parameters here. Pruning has little effect on the accuracy, usually the accuracy decreases. The default value of j-48 often performs better, but I tried to change confidenceFactor and got a slight increase in F-Measure.