Comp309 — Machine Learning Tools and Techniques
Assignment1: Sprint on the dataset each

Part1: Investigate basic use of five tribe of AI

Heart Discuss Description:
AI tools Used: Weka, Keel
Relation: **Heart Discuss Cleveland**
Instances: 303
Attributes: 14
- age
- sex
- cp
- trestbps
- chol
- fbs
- restecg
- thalach
- exang
- oldpeak
- slope
- ca
- thal
- num

Test mode: 10-fold cross-validation

The "goal" field refers to the presence of heart disease in the patient. It is integer valued from 0 (no presence) to 4. Experiments with the Cleveland database have concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

1.1
Describe the results of each technique on the one dataset. Note the most appropriate form of results may differ between each technique. Exercise your skill and judgement to decide how the results should be communicated. Note, it is the description that is important here, not the experimental method, e.g. setting up training and test sets properly is explored in Part 2.

Symbolists:
- **Decision tree J48**

Correctly Classified Instances	165	54.4554 %
Incorrectly Classified Instances	138	45.5446 %
Kappa statistic	0.2034	
Mean absolute error	0.2138	
Root mean squared error	0.3483	
Relative absolute error	82.5114 %	
Root relative squared error	96.9573 %	
Total Number of Instances	303	

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.866	0.396	0.721	0.866	0.787	0.491	0.746	0.699	0
	0.091	0.149	0.119	0.091	0.103	-0.065	0.446	0.165	1
	0.139	0.086	0.179	0.139	0.156	0.059	0.541	0.157	2
	0.200	0.063	0.292	0.200	0.237	0.162	0.545	0.167	3
	0.077	0.038	0.083	0.077	0.080	0.041	0.444	0.049	4
Weighted Avg.	0.528	0.260	0.470	0.528	0.494	0.281	0.631	0.448	

Connectionists:

- multilayerPerceptron

Correctly Classified Instances	162	53.4653 %
Incorrectly Classified Instances	141	46.5347 %
Kappa statistic	0.2669	
Mean absolute error	0.1904	
Root mean squared error	0.3878	
Relative absolute error	73.4822 %	
Root relative squared error	107.9639 %	
Total Number of Instances	303	

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.823	0.259	0.789	0.823	0.806	0.567	0.859	0.854	0
	0.182	0.177	0.185	0.182	0.183	0.004	0.580	0.224	1
	0.194	0.086	0.233	0.194	0.212	0.117	0.711	0.224	2
	0.257	0.104	0.243	0.257	0.250	0.149	0.746	0.290	3
	0.077	0.034	0.091	0.077	0.083	0.046	0.684	0.082	4
Weighted Avg.	0.535	0.196	0.521	0.535	0.527	0.341	0.770	0.566	

Evolutionaries:

TEST RESULTS

==========

Classifier= cleveland

Fold 0 : CORRECT=0.6333333333333333 N/C=0.0

Fold 1 : CORRECT=0.6 N/C=0.0

Fold 2 : CORRECT=0.5666666666666667 N/C=0.0

Fold 3 : CORRECT=0.6 N/C=0.0

Fold 4 : CORRECT=0.6333333333333333 N/C=0.0

Fold 5 : CORRECT=0.4 N/C=0.0

Fold 6 : CORRECT=0.5806451612903225 N/C=0.0

Fold 7 : CORRECT=0.5483870967741935 N/C=0.0

Fold 8 : CORRECT=0.6 N/C=0.0

Fold 9 : CORRECT=0.6451612903225806 N/C=0.0

Global Classification Error + N/C:

0.419247311827957

stddev Global Classification Error + N/C:

0.06686313402150826

Correctly classified:

0.5807526881720431

Global N/C:

0.0

TRAIN RESULTS

==========

Classifier= cleveland

Summary of data, Classifiers: cleveland

Fold 0 : CORRECT=0.9963369963369964 N/C=0.0

Fold 1 : CORRECT=0.9523809523809523 N/C=0.0

Fold 2 : CORRECT=0.9706959706959707 N/C=0.0

Fold 3 : CORRECT=0.9633699633699634 N/C=0.0

Fold 4 : CORRECT=0.9706959706959707 N/C=0.0

Fold 5 : CORRECT=0.9597069597069597 N/C=0.0

Fold 6 : CORRECT=0.9632352941176471 N/C=0.0

Fold 7 : CORRECT=0.9632352941176471 N/C=0.0

Fold 8 : CORRECT=0.9706959706959707 N/C=0.0

Fold 9 : CORRECT=0.9522058823529411 N/C=0.0

Global Classification Error + N/C:
0.03374407455289809
stddev Global Classification Error + N/C:
0.011930622684865253
Correctly classified:
0.9662559254471019
Global N/C:
0.0

Bayesian:

- Naive bayes

Correctly Classified Instances	169	55.7756 %
Incorrectly Classified Instances	134	44.2244 %
Kappa statistic	0.305	
Mean absolute error	0.1843	
Root mean squared error	0.3371	
Relative absolute error	71.1406 %	
Root relative squared error	93.8447 %	
Total Number of Instances	303	

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.854	0.209	0.828	0.854	0.841	0.647	0.896	0.909	0
	0.200	0.177	0.200	0.200	0.200	0.023	0.651	0.245	1
	0.222	0.112	0.211	0.222	0.216	0.107	0.775	0.233	2
	0.257	0.093	0.265	0.257	0.261	0.166	0.805	0.271	3
	0.077	0.021	0.143	0.077	0.100	0.076	0.792	0.113	4
Weighted Avg.	0.558	0.170	0.546	0.558	0.552	0.390	0.822	0.600	

Analogises-

- K nearest neighbour (lazy IBK)

Correctly Classified Instances	168	55.4455 %
Incorrectly Classified Instances	135	44.5545 %
Kappa statistic	0.2991	
Mean absolute error	0.1808	
Root mean squared error	0.4184	
Relative absolute error	69.7703 %	
Root relative squared error	116.4767 %	
Total Number of Instances	303	

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.811	0.266	0.782	0.811	0.796	0.547	0.786	0.756	0
	0.291	0.157	0.291	0.291	0.291	0.134	0.584	0.231	1
	0.250	0.094	0.265	0.250	0.257	0.160	0.555	0.157	2
	0.257	0.093	0.265	0.257	0.261	0.166	0.574	0.155	3
	0.077	0.031	0.100	0.077	0.087	0.052	0.612	0.061	4
Weighted Avg.	0.554	0.196	0.543	0.554	0.548	0.361	0.690	0.490	

The data above is the results of all the implemented algorithms, it presents the accuracy of implemented algorithms on the given heart disease dataset, the lowest accuracy is 53.4653 % it presents by multiple perceptron algorithms, and the highest accuracy is 55.7756 % presents by Naive Bayes algorithms. I have been used the 10-fold cross-validation as the test mode, so the test result is quite fair. The result shows that the Naive Bayes algorithms slightly have better performance than other implemented algorithms. But if acting in the real world, the 55.44% accuracy is not that satisfying. If we look at the rate of true positives (TP rate) of all implemented algorithms, they have the same commonality: to correctly classified class 0 is very high. So, it has high accuracy to correctly classifier presence of heart disease in the patient (0 means no heart disease). Therefore which means is better to use machine learning tools to classify the presence of heart disease in the patient rather than the heart disease level.

1.2 The report should detail why each selected technique from the stated tool belongs to a given tribe, i.e. identify the important aspects of the technique in terms of

Symbolist is focussing on the premise of inverse deduction. Instead of the classical model of starting with a premise and looking for the conclusions, inverse deduction starts with a set of premises and conclusions and works backward to fill in the gaps. J48 is a type of Decision tree algorithm that according to the given instance and class labels to build a decision tree. For the decision tree structures, leaves represent class labels and branches represent conjunctions of features that lead to the class labels. Once the decision tree has been built, the user can use the instance's attributes to chase down on diction tree to which is to find out the class label. That is an inverse deduction to work out the logic for classifier the presence of heart disease in the patient. And the evaluation method it uses is accuracy, that is the percentage of correct in the test. Therefore J48 belongs to the Symbolists tribes.

Connectionists is based on connecting artificial neutrons in a neural network, to re-engineer the human brain. The technique I use above is multilayer Perceptron that is a classifier that uses backpropagation to classify instances. And the backpropagation is a method used in artificial neural networks to calculate a gradient that is needed in the calculation of the weights to be used in the network. It uses a gradient descent method involves calculating the squared error function with respect to the weights of the network. A common evaluation method of backpropagation uses the mean-squared error, which tries to minimize the average squared error between the network's output, and the target value over all the example pairs. It will according to the square error to change the current weights, and the weights are the connections between two artificial neutrons. Therefore multilayer perceptron belongs to Connectionists.

Evolutionaries is an algorithms that will constantly evolve and adapt to unknown conditions and processes. The technique I used above is Genetic Algorithm it represents Genetic programming, is to simulated the process of biological evolution to generate a algorithms that can find out the solutions. The optimisation driver is genetic search . It begins with a group of randomly generate population Then, according to the ability to complete a given task to determine the fitness of the algorithms. From the best fit algorithms, the computer program will simulate between combination, mutation, gene replication, gene deletion and other generations, until achieve predetermined a stop condition. Therefore Genetic Algorithm is belongs to the Evolutionaries.

Bayesian is an algorithm that will constantly evolve and adapt to unknown conditions and processes. The technique I used above is Genetic Algorithm it represents Genetic programming, is to simulate the process of biological evolution to generate an algorithm that can find out the solutions. The optimization driver is a genetic search. It begins with a group of randomly generated population Then, according to the ability to complete a given task to determine the fitness of the algorithms. From the best-fit algorithms, the computer program will simulate between combination, mutation, gene replication, gene deletion and other generations, until achieve predetermined a stop condition. Therefore Genetic Algorithm belongs to the Evolutionaries.

Analogises are focused on techniques to match bits of data to each other, the representation of analogies is the support vector which can give results to neural network models.
The technique I selected from the analogies is the lazy IBK, lazy IBK is a type of K nearest neighbor technique. That is a nonparametric statistical method for classification and regression, which is the same as support vectors. When input contains the k nearest feature space training sample, accordance the evaluation method margin to calculate K nearest neighbor, pick the most class label form K nearest neighbor as the class label for this training sample. Therefore lazy IBK belongs to the analogies.

1.3 Identify how these aspects of the technique are different in relation to the dataset:

They are two types of data in the heart disease dataset, one is numeric types eg age [0,1,2,3,4] another one is normal type eg sex [male, female].

In the J48 (Decision tree) the main idea is to classier the data by nominal type, but the attributes of Heart disease dataset is mix with numeric and nominal type. J48 did convert numeric to nominal by a critical point, that split the data half. By this way cannot be exactly classifier the Heart disease dataset, so J48 tress is not a good classifier technique.

In the naive Bayes technique, it assumes all the attributes are independent of each other. In the Heart disease dataset, most of all attributes are independent but, there have a numeric type dataset which it does not make sense if we use naive Bayes technique to separate them. Naive Bayes is not a good classifier technique either.

In the Genetic Algorithm, the input of the attributes should be a numeric type, otherwise is hard to design the fitness function. In my heart disease data, most of the attributes are the nominal type. If we remove all the nominal type attributes, to do the classification the result will become not accurate.

In the multilayer Perceptron, it can be dealing with binary input and numeric inputs. If I input the data without doing data preparation, the input nominal type attributes will become meaningless.

In the K-nearest neighbor (lazy IBK), is meaningless when the classifier treats the numberless type as the numeric type, that will effect on the result.

(optional) Insight into ‘ why one or more aspect is suited to a given dataset ’ will be needed to achieve high grades.

J48 is suited with nominal type data and binary data (eg 1 or 0). Because it can split the attributes into several branches, which adapted to the tree structure. If it used the numeric type attributes to build the decision tree it will lead to having infinite branches. If there is a new instance with unseen numeric attributes come to the diction tree, that will become no way to classier.

Naive Bayes is the same idea as J48.

Genetic Algorithm is suited with numeric data. Because normally the fitness function is using arithmetic operations. Is hard to design the fitness function if using are the type of dataset eg (nominal type).

Multilayer Perceptron is suited to binary and numeric data. It referent the backpropagation algorithm to change weights. The adjust weights algorithm did use arithmetic operations. In another world, use any other type of attributes did not work for multilayer Perceptron method.

K-nearest neighbor is suited with numeric data. Because in K-nearest neighbor method, the mean idea is to calculate the distance between testing instance to any other instance in the training dataset to find out the nearest neighbors. To calculate the distance, we have to use the following algorithm :

$$dist\,(A, B) = \sqrt{\frac{\sum_{i=1}^{m} (x_i - y_i)^2}{m}}$$

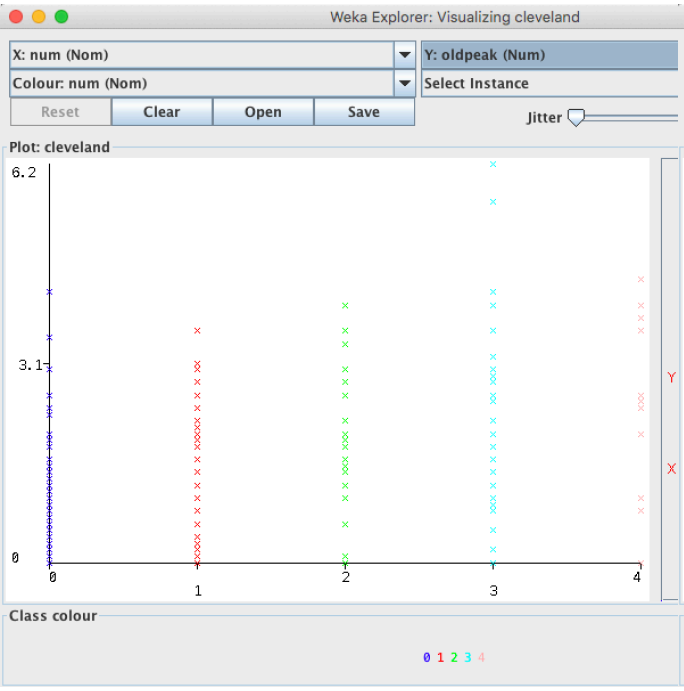
Therefore we have to use the numbers to do the calculation, otherwise is hard to judge the distances.

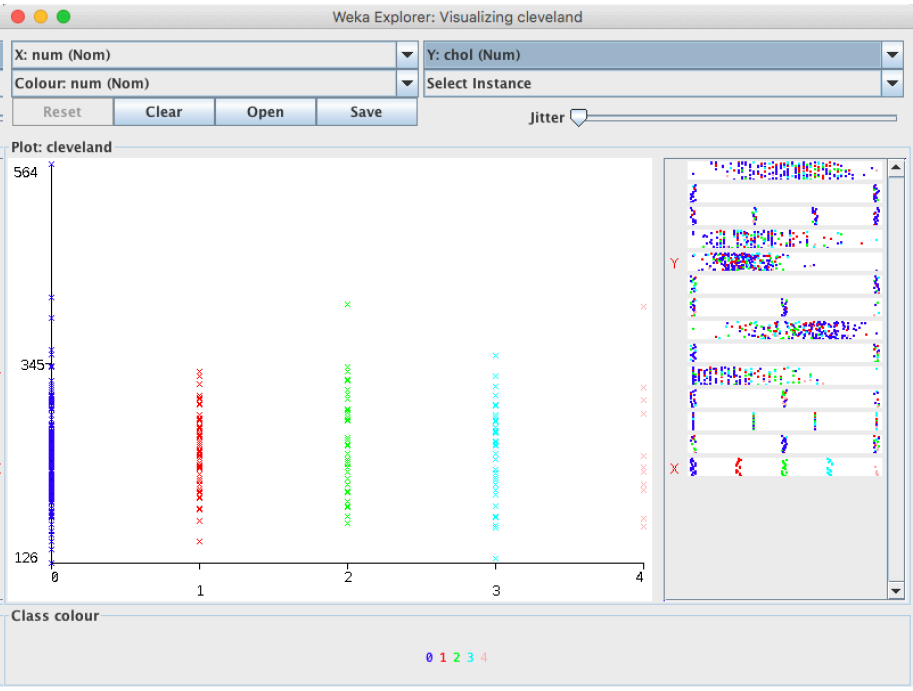
Part 2 consider a pipeline for dataset processing

Pipeline
Data Pipeline is a system throughout the entire product or data, and the data is the main object of this Pipeline carries. Data Pipeline connects the different Data processing analysis of each link, make the entire system becomes confused sleek, manageable and extensions.

Business understanding:
Consider the business aspects of the heart disease dataset, the acquisition hope to achieve is to predict the presence of heart disease in the patient.

Data understanding:
In the aspects of data understanding, we have to collect the data of patients, but we do not know which attribute as a direct result of proving the presence of heart disease. So, we start with using a huge amount of attribute for training then gradually reducing the attribute that has less impact for heart disease. In the original databases, there have 76 raw attributes, but only 14 of them are actually used. And there are two types of the data nominal type and numeric types. Also, there have several missing attributes values Eg number of major vessels and Thal. From my personal observation, there are some outlier values in oldPeak attributes and chol attributes. That will have an impact on normalizing the data.





Data preparation:
In data preparation it will focus on the Data preparation stage includes never processing of data structure in the final data set of all activities. These data will be the modeling phase of the input values, the task including the selection of attribute extraction, data tables, records, and transform the data format to required by the model tools and cleaning data.

Modeling:
In the aspects of data modelling, this pipeline could suit Symbolists and Bayesian tribes of AI

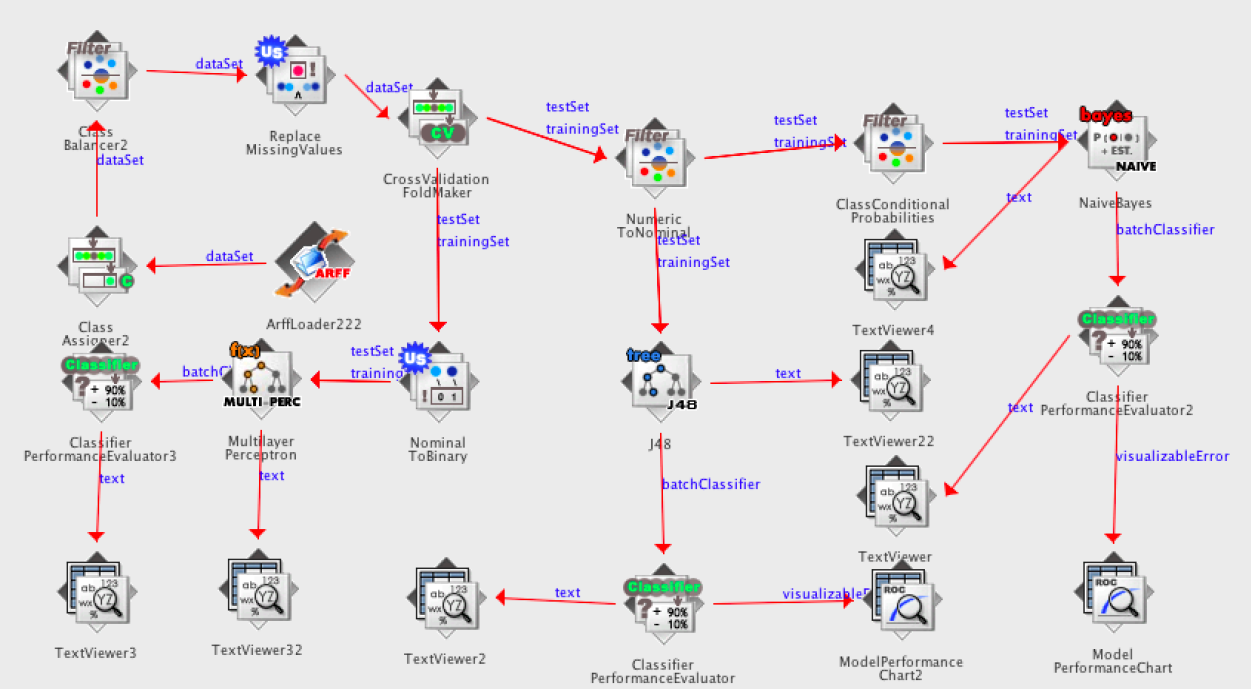
Evaluation:
In the aspects of evaluation, this pipeline supports Decision tree and Naive Bayes method to evaluate a solution

Deployment:
In the aspects of Deployment, I plan to combine all the heart discuss levels to 1, which means the to classify the presence of heart disease in the patient rather than the heart disease level. Because that is the gold of this data set, and easier to classify, if we want to check whatever has a different level of the heart disease. Is better to do one vs one, eg (have heart disease vs not have heart disease) rather than one to many. Also, normalize all the numeric type data, unify them in the same type, remove the noisy, outlier and missing data instances. calculate a critical point to divide them. On the other hand is an import that we divide the training set and test set evenly. If the scale is 7:3 no Heart disease vs Heart disease in the training set, it must have the same scale 7:3 in the test set.

K-fold cross-validation:
K-fold cross-validation is one of the test methods to test the performance of the algorithm. The procedure is followed:
1 Split the dataset into k small dataset
2 For each unique small dataset, take one small dataset as the test set and the rest are training set.
3 train k times each time are use different test set, make sure each small dataset have once to be the test set.
4 average all the training test accuracy as the final result.

Part 3: Use the pipeline to revaluate the five tribes of AI to classify the dataset

3.1 pipeline:



Before I use the heart disease dataset run through this pipeline, I did combine the different level of heart disease classes into one class. Because I can not find this function in Weka, so I have done it with Excel. Furthermore, through the pipeline, I make the class balance and replace all the missing values with the means from the training data. And Before doing the classification I convert all the attribute type to suitable type for the techniques.

Compare and contrast the results between the different tools.

Is hard to say the differ or compare the results between the different tool through my pipeline. Because before doing the classification, they all did the similar data preparation. The only difference is to convert all the attribute type to suitable type. I can not say that

multilayer Perceptron has better improve on accuracy (26%) than J48 (21%), because multilayer Perceptron did the better conversion.

3.2.1 Accuracy in terms of the fraction of the test instances that it classified correctly.

Bayesian (Naive Bayes):

=== Evaluation result ===

Correctly Classified Instances	246	81.1881 %
Incorrectly Classified Instances	57	18.8119 %
Kappa statistic	0.6206	
Mean absolute error	0.1922	
Root mean squared error	0.3928	
Relative absolute error	38.6913 %	
Root relative squared error	78.8242 %	
Total Number of Instances	303	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.835	0.216	0.820	0.835	0.828	0.621	0.883	0.888	0
	0.784	0.165	0.801	0.784	0.793	0.621	0.883	0.849	1
Weighted Avg.	0.812	0.192	0.812	0.812	0.812	0.621	0.883	0.870	

=== Confusion Matrix ===

a	b	<-- classified as
137	27	a = 0
30	109	b = 1

Symbolists (J48):

=== Evaluation result ===

Correctly Classified Instances	228	75.2475 %
Incorrectly Classified Instances	75	24.7525 %
Kappa statistic	0.5029	
Mean absolute error	0.3066	
Root mean squared error	0.435	
Relative absolute error	61.7352 %	
Root relative squared error	87.3014 %	
Total Number of Instances	303	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.756	0.252	0.780	0.756	0.768	0.503	0.783	0.753	0
	0.748	0.244	0.722	0.748	0.735	0.503	0.783	0.722	1
Weighted Avg.	0.752	0.248	0.753	0.752	0.753	0.503	0.783	0.738	

=== Confusion Matrix ===

```
  a   b   <-- classified as
 124  40 |   a = 0
  35 104 |   b = 1
```

Connectionists (Multilayer Perceptron):

```
=== Evaluation result ===

Correctly Classified Instances         245.8585        81.1414 %
Incorrectly Classified Instances        57.1415        18.8586 %
Kappa statistic                          0.6228
Mean absolute error                      0.2017
Root mean squared error                  0.4175
Relative absolute error                 40.3279 %
Root relative squared error             83.5025 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.817    0.194    0.808      0.817    0.812      0.623    0.870     0.875     0
               0.806    0.183    0.815      0.806    0.810      0.623    0.870     0.858     1
Weighted Avg.  0.811    0.189    0.811      0.811    0.811      0.623    0.870     0.866

=== Confusion Matrix ===

   a      b      <-- classified as
 123.79  27.71 |      a = 0
  29.43 122.07 |      b = 1
```

3.2.2 . Report a snapshot of the learned classifier s discovered by your program.

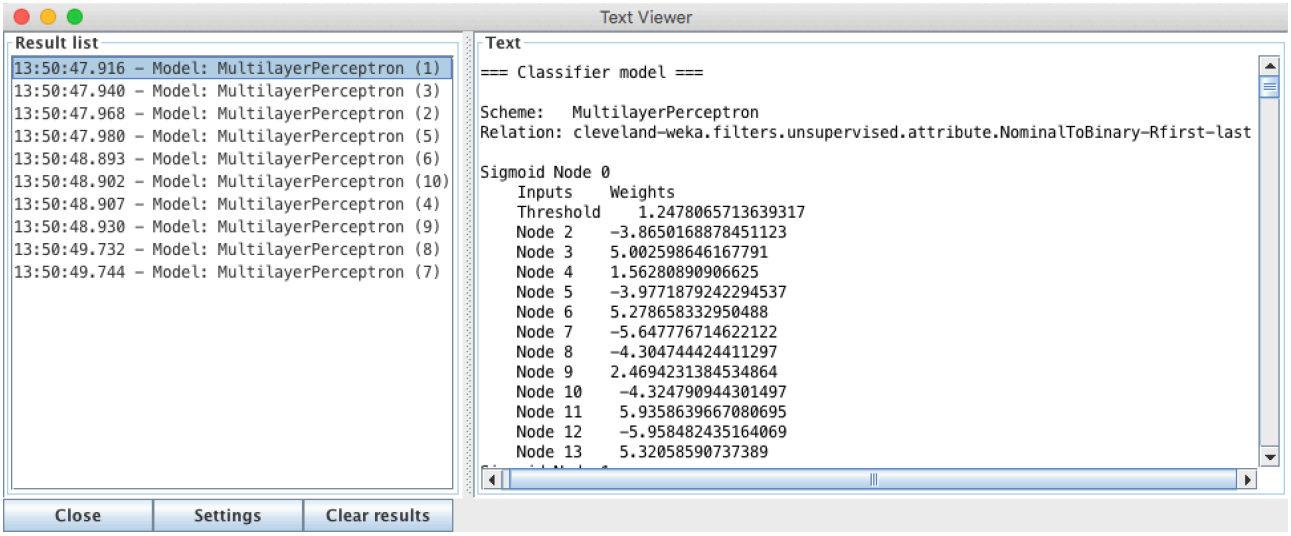
```
=== Classifier model ===
```

Scheme: J48

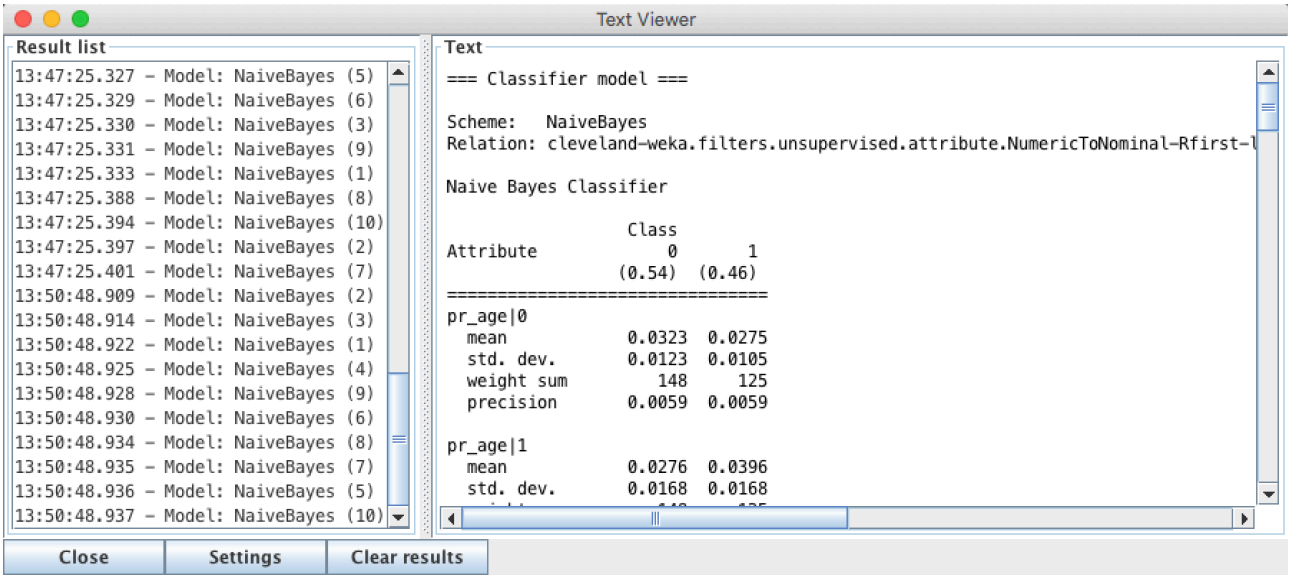
```
J48 pruned tree
------------------

thal = 6
|   ca = 0: 0 (7.13/1.06)
|   ca = 1: 1 (4.0)
|   ca = 2: 1 (4.0)
|   ca = 3: 1 (2.0)
thal = 3
|   ca = 0: 0 (108.54/12.56)
|   ca = 1
|   |   sex = 0: 0 (12.0)
|   |   sex = 1
|   |   |   cp = 1: 0 (2.0/1.0)
```

| | | cp = 4: 1 (8.0)
| | | cp = 3: 0 (2.34)
| | | cp = 2: 0 (2.0/1.0)
| ca = 2
| | restecg = 2: 1 (4.0)
| | restecg = 0: 0 (8.16/3.0)
| | restecg = 1: 1 (0.0)
| ca = 3: 1 (6.08/1.08)
thal = 7: 1 (102.75/22.38)
Number of Leaves : 15
Size of the tree : 21

Scheme Multiple Perceptron:



Scheme Naive bayes:



3.2.3 Compare the accuracy of your techniques before and after using a data pipeline approach. Please comment on any difference s, suggesting reasons .

Compare the accuracy of my techniques before and after: using a data pipeline approach, the accuracy of all my techniques (Naive Bayes, J48, Multilayer Perceptron) are improved about 20%. But I did not implement the evolutionary and k-nearest neighbor method, the nominal type

value just not work with those methods. The reason increased the accuracy is I combine the different level of heart disease classes into one class. That makes the data have balance class and become much easier to classier. Otherwise, one error classifies in the small class will decrease larger accuracy. Before I run through the multilayer perceptron classifier, I transfer the nominal to binary attribute, and before I run through the J48 and naive Bayes classifier I transfer the numerical to nominal attribute. That will become meaningful during the classification. Thus the result shows it is helpful when I do those changes.