

MC0717

DATA MINING LAB MANUAL

Index

S.No	Experiment	Page no	Signature
1.	Demonstration of preprocessing on dataset student.arff		
2.	Demonstration of preprocessing on dataset labor.arff		
3.	Demonstration of Association rule process on dataset contactlenses.arff using apriori algorithm		
4.	Demonstration of Association rule process on dataset test.arff using apriori algorithm		
5.	Demonstration of classification rule process on dataset student.arff using j48 algorithm		
6.	Demonstration of classification rule process on dataset employee.arff using j48 algorithm		
7.	Demonstration of classification rule process on dataset employee.arff using id3 algorithm		
8.	Demonstration of classification rule process on dataset employee.arff using naïve bayes algorithm		
9.	Demonstration of clustering rule process on dataset iris.arff using simple k-means		
10.	Demonstration of clustering rule process on dataset student.arff using simple k means		

1. Demonstration of preprocessing on dataset student.arff

Aim: This experiment illustrates some of the basic data preprocessing operations that can be performed using WEKA-Explorer. The sample dataset used for this example is the student data available in arff format.

Step1: Loading the data. We can load the dataset into weka by clicking on open button in preprocessing interface and selecting the appropriate file.

Step2: Once the data is loaded, weka will recognize the attributes and during the scan of the data weka will compute some basic statistics on each attribute. The left panel in the above figure shows the list of recognized attributes while the top panel indicates the names of the base relation or table and the current working relation (which are same initially).

Step3: Clicking on an attribute in the left panel will show the basic statistics on the attributes for the categorical attributes the frequency of each attribute value is shown, while for continuous attributes we can obtain min, max, mean, standard deviation and deviation etc.,

Step4: The visualization in the right button panel in the form of cross-tabulation across two attributes.

Note: we can select another attribute using the dropdown list.

Step5: Selecting or filtering attributes

Removing an attribute-When we need to remove an attribute, we can do this by using the attribute filters in weka. In the filter model panel, click on choose button, This will show a popup window with a list of available filters.

Scroll down the list and select the “weka.filters.unsupervised.attribute.remove” filters.

Step 6: a) Next click the textbox immediately to the right of the choose button. In the resulting dialog box enter the index of the attribute to be filtered out.

b) Make sure that invert selection option is set to false. Click OK now in the filter box. You will see “Remove-R-7”.

c) Click the apply button to apply filter to this data. This will remove the attribute and create new working relation.

d) Save the new working relation as an arff file by clicking save button on the top(button) panel. (student.arff)

Discretization

1) Sometimes association rule mining can only be performed on categorical data. This requires performing discretization on numeric or continuous attributes. In the following example let us discretize age attribute.

Let us divide the values of age attribute into three bins(intervals).

First load the dataset into weka(student.arff)

Select the age attribute.

Activate filter-dialog box and select “WEKA.filters.unsupervised.attribute.discretize” from

the list.

To change the defaults for the filters, click on the box immediately to the right of the choose button.

We enter the index for the attribute to be discretized. In this case the attribute is age. So we must enter '1' corresponding to the age attribute.

Enter '3' as the number of bins. Leave the remaining field values as they are.

Click OK button.

Click apply in the filter panel. This will result in a new working relation with the selected attribute partitioned into 3 bins.

Save the new working relation in a file called student-data-discretized.arff

Dataset student .arff

@relation student

@attribute age {<30,30-40,>40}

@attribute income {low, medium, high}

@attribute student {yes, no}

@attribute credit-rating {fair, excellent}

@attribute buyspc {yes, no}

@data

%

<30, high, no, fair, no

<30, high, no, excellent, no

30-40, high, no, fair, yes

>40, medium, no, fair, yes

>40, low, yes, fair, yes

>40, low, yes, excellent, no

30-40, low, yes, excellent, yes

<30, medium, no, fair, no

<30, low, yes, fair, no

>40, medium, yes, fair, yes <30,

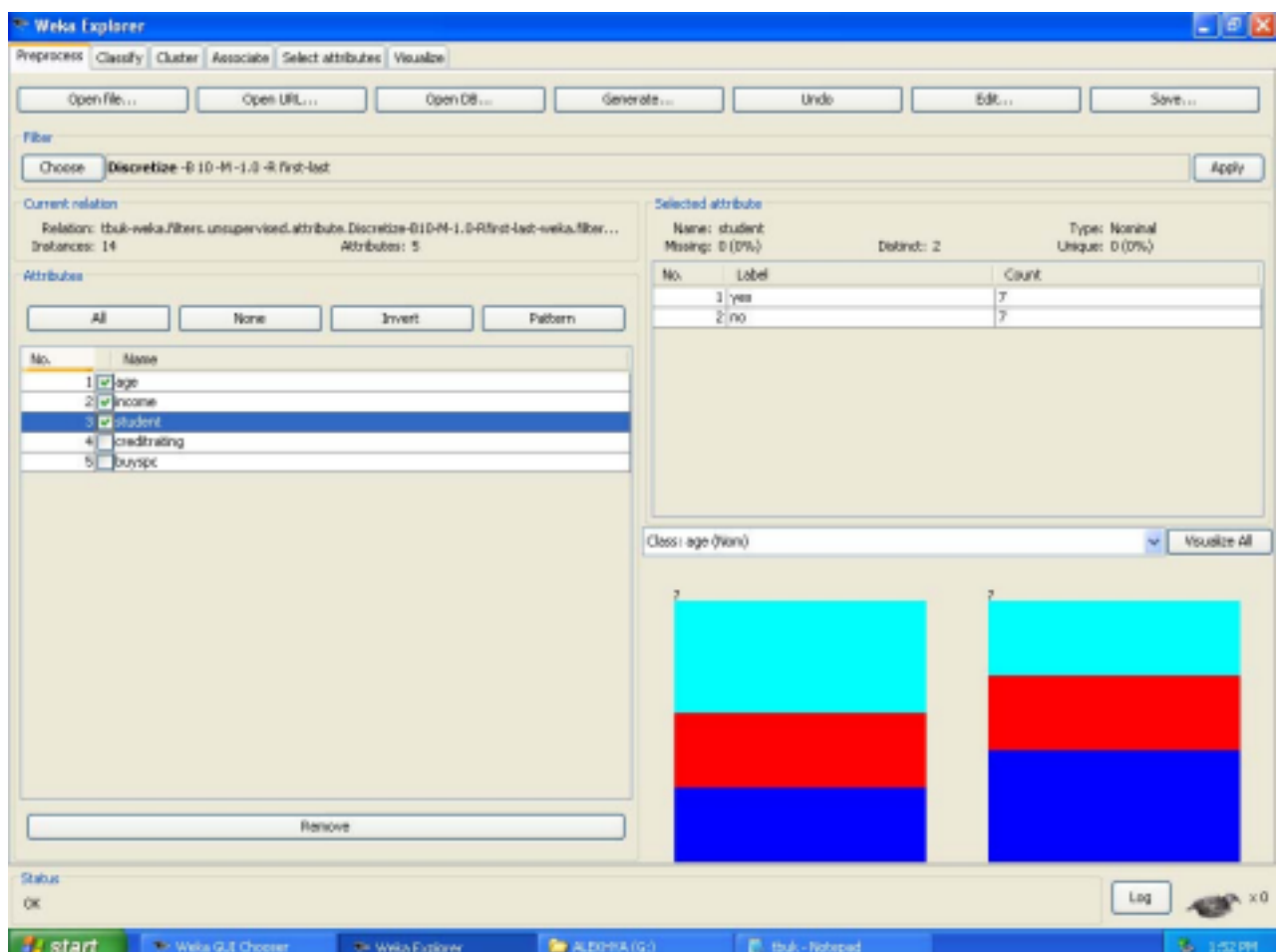
medium, yes, excellent, yes

30-40, medium, no, excellent, yes

30-40, high, yes, fair, yes >40,

medium, no, excellent, no %

The following screenshot shows the effect of discretization.



2. Demonstration of preprocessing on dataset labor.arff

Aim: This experiment illustrates some of the basic data preprocessing operations that can be performed using WEKA-Explorer. The sample dataset used for this example is the labor data available in arff format.

Step1: Loading the data. We can load the dataset into weka by clicking on open button in preprocessing interface and selecting the appropriate file.

Step2: Once the data is loaded, weka will recognize the attributes and during the scan of the data weka will compute some basic statistics on each attribute. The left panel in the above figure shows the list of recognized attributes while the top panel indicates the names of the base relation or table and the current working relation (which are same initially).

Step3: Clicking on an attribute in the left panel will show the basic statistics on the attributes for the categorical attributes the frequency of each attribute value is shown, while for continuous attributes we can obtain min, max, mean, standard deviation and deviation etc.,

Step4: The visualization in the right button panel in the form of cross-tabulation across two attributes.

Note: we can select another attribute using the dropdown list.

Step5: Selecting or filtering attributes

Removing an attribute-When we need to remove an attribute, we can do this by using the attribute filters in weka. In the filter model panel, click on choose button, This will show a popup window with a list of available filters.

Scroll down the list and select the “weka.filters.unsupervised.attribute.remove” filters.

Step 6:a) Next click the textbox immediately to the right of the choose button. In the resulting dialog box enter the index of the attribute to be filtered out.

b) Make sure that invert selection option is set to false. Click OK now in the filter box. You will see “Remove-R-7”.

c) Click the apply button to apply filter to this data. This will remove the attribute and create new working relation.

d) Save the new working relation as an arff file by clicking save button on the top(button)panel.(labor.arff)

Discretization

1) Sometimes association rule mining can only be performed on categorical data. This requires performing discretization on numeric or continuous attributes. In the following example let us discretize duration attribute.

Let us divide the values of duration attribute into three bins(intervals).

First load the dataset into weka(labor.arff)

Select the duration attribute.

Activate filter-dialog box and select “WEKA.filters.unsupervised.attribute.discretize” from the list.

To change the defaults for the filters, click on the box immediately to the right of the choose button.

We enter the index for the attribute to be discretized. In this case the attribute is duration So we must enter ‘1’ corresponding to the duration attribute.

Enter ‘1’ as the number of bins. Leave the remaining field values as they are.

Click OK button.

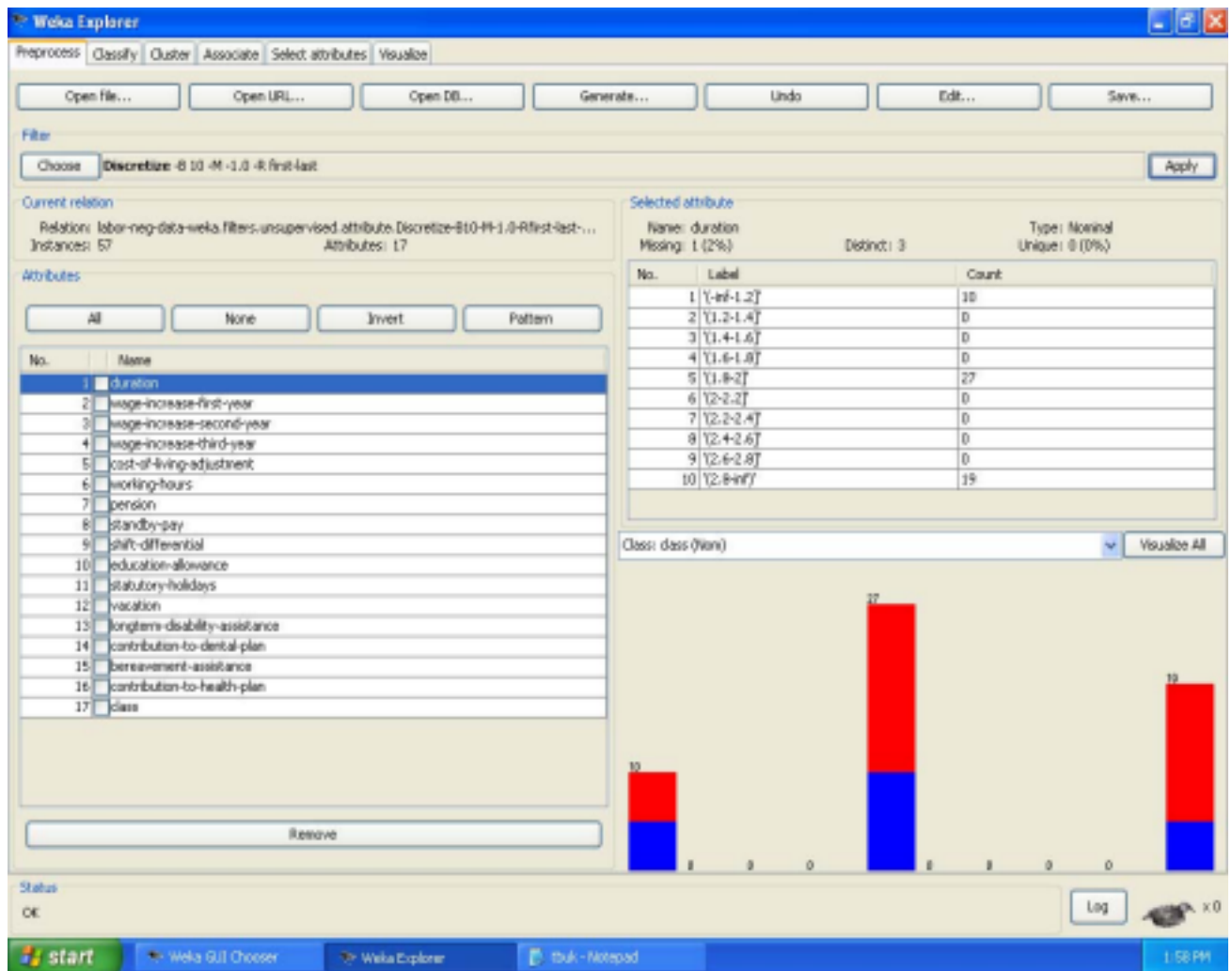
Click apply in the filter panel. This will result in a new working relation with the selected attribute partition into 1 bin.

Save the new working relation in a file called labor-data-discretized.arff

Dataset labor.arff

No.	duration	wage-increase-first-year	wage-increase-second-year	wage-increase-third-year	cost-of-living-adjustment	working-hours	pension	standby-pay	shift-differential	educ
1	3.0	5.0				40.0			2.0	
2	2.0	4.5	5.0			35.0	net_aka			yes
3						30.0	empl_s...		5.0	
4	3.0	3.7	4.0	5.0	bc					yes
5	3.0	4.5	4.5	5.0		40.0				
6	2.0	2.0	2.5			35.0			6.0	yes
7	3.0	4.0	5.0	5.0	bc		empl_s...			
8	3.0	6.9	4.0	2.3		40.0			3.0	
9	2.0	3.0	7.0			30.0		12.0	25.0	yes
10	3.0	5.7			none	40.0	empl_s...		4.0	
11	3.0	3.5	4.0	4.5	none	30.0			3.0	
12	2.0	6.4	6.4			30.0			4.0	
13	2.0	3.5	4.0		none	40.0			2.0	no
14	3.0	3.5	4.0	5.1	bcd	37.0			4.0	
15	3.0	3.0			none	36.0			10.0	no
16	2.0	4.5	4.0		none	37.0	empl_s...			
17	3.0	2.0				35.0			2.0	
18	3.0	2.1			bc	40.0	net_aka	2.0	3.0	no
19	3.0	2.0			none	30.0	none			yes
20	2.0	4.0	5.0		bcd	35.0		13.0	5.0	
21	2.0	4.3	4.4			30.0			4.0	
22	2.0	2.5	3.0			40.0	none			
23	3.0	3.5	4.0	4.5	bcd	27.0				
24	2.0	4.5	4.0			40.0			4.0	
25	3.0	6.0				30.0		0.0	3.0	
26	3.0	2.0	2.0	2.0	none	40.0	none			
27	2.0	4.5	4.5		bcd					yes
28	2.0	3.0	3.0		none	33.0				yes
29	2.0	5.0	4.0		none	37.0			5.0	no
30	3.0	2.0	2.5			35.0	none			
31	3.0	4.5	4.5	5.0	none	40.0				no
32	3.0	3.0	2.0	3.5	bc	40.0	none		5.0	no
33	2.0	2.5	2.5			30.0	empl_s...			
34	2.0	4.0	5.0		none	40.0	none		3.0	no
35	3.0	2.0	2.5	2.1	bc	40.0	none	2.0	1.0	no
36	2.0	2.0	2.0		none	40.0	none			no
37	3.0	2.0			bc	40.0	net_aka	4.0	6.0	no
38	3.0	2.0			none	30.0	empl_s...	2.0	3.0	no

The following screenshot shows the effect of discretization



3. Demonstration of Association rule process on dataset contactlenses.arff using apriori algorithm

Aim: This experiment illustrates some of the basic elements of association rule mining using WEKA. The sample dataset used for this example is contactlenses.arff

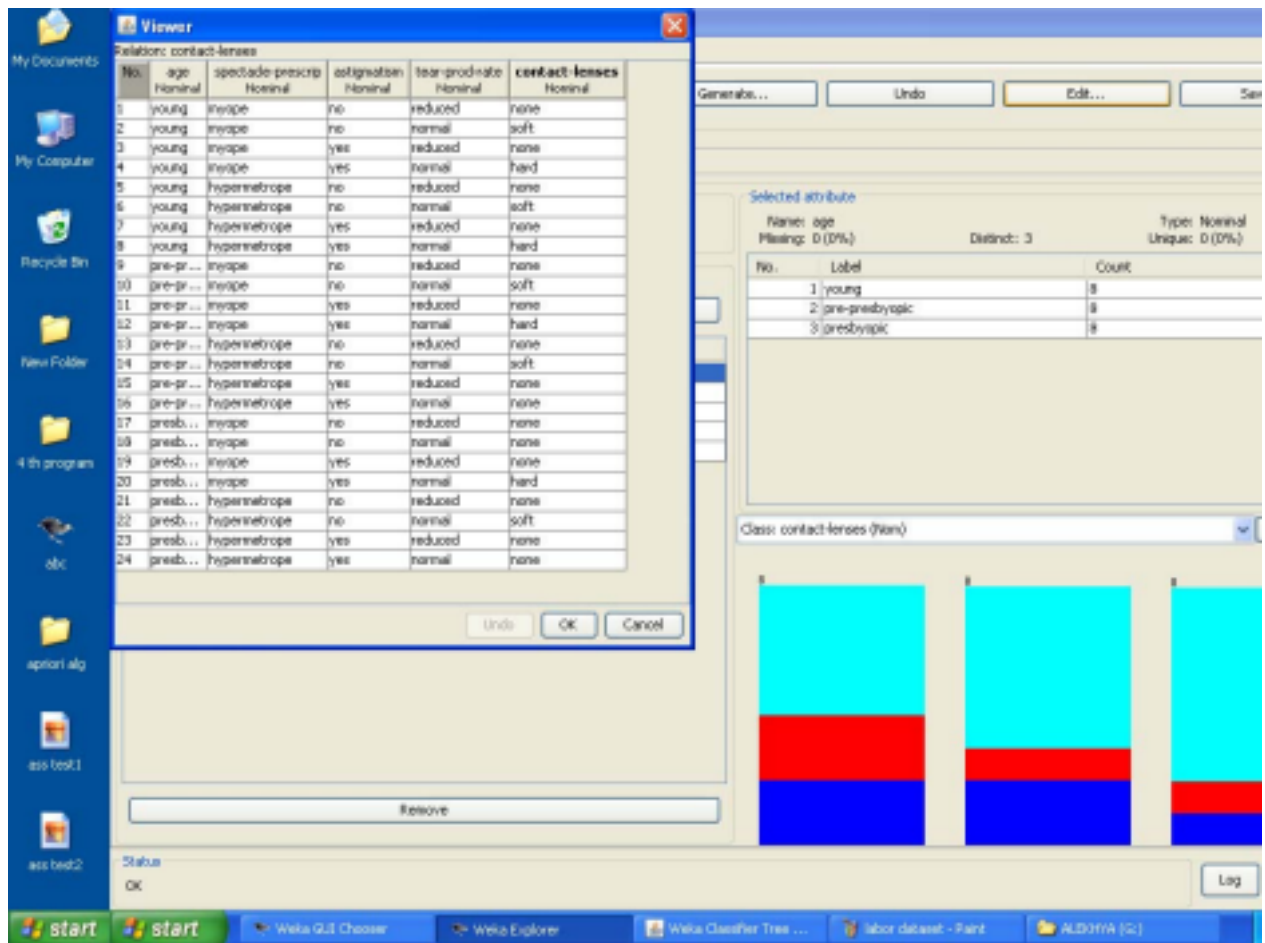
Step1: Open the data file in Weka Explorer. It is presumed that the required data fields have been discretized. In this example it is age attribute.

Step2: Clicking on the associate tab will bring up the interface for association rule algorithm.

Step3: We will use apriori algorithm. This is the default algorithm.

Step4: Inorder to change the parameters for the run (example support, confidence etc) we click on the text box immediately to the right of the choose button.

Dataset contactlenses.arff



The following screenshot shows the association rules that were generated when apriori algorithm is applied on the given dataset.

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Cluster

Choose: **SimpleKMeans-W2-5-18**

Cluster mode: ☒ Use training set ☐ Randomised test set

Cluster output: ***** RRM INFORMATION *****

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Associate

Choose: **Apriori-M18-T0-C0.9-D0.05-U18-M0.1-S-1.0**

Start | Stop

Result list (right click for):

- 12/09/06 - Apriori

Associate output:

```

*** RRM information ***

Scheme:      weka.associations.Apriori -M 18 -T 0 -C 0.9 -D 0.05 -U 18 -M 0.1 -S -1.0
Relation:    contact-lenses
Instances:    24
Attributes:   5
              age
              spectacle-prescrip
              astigmatism
              tear-prod-rate
              contact-lenses

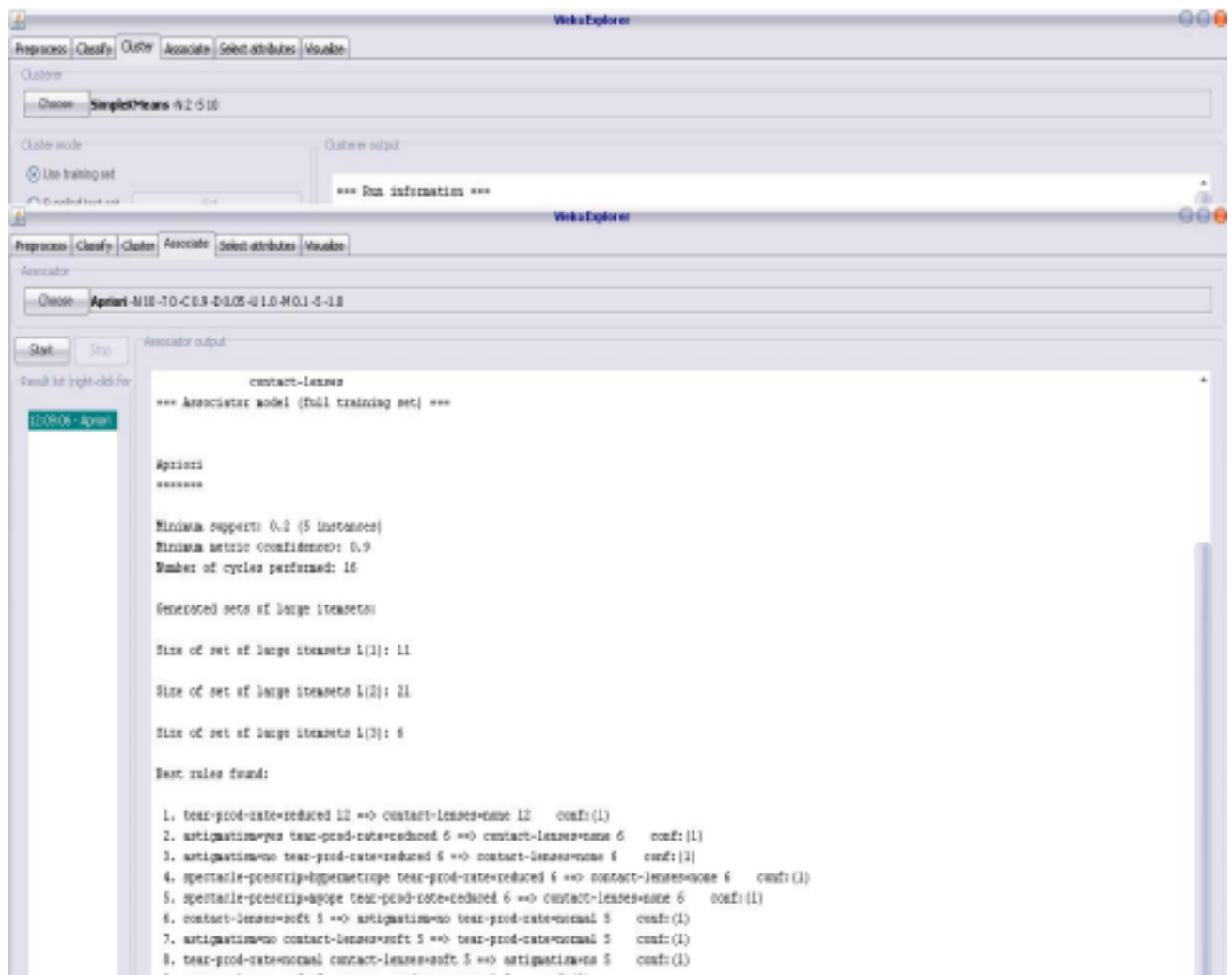
*** Associator model (full training set) ***

Apriori
*****

Minimum support: 0.2 (5 instances)
Minimum metric (confidence): 0.9
Number of cycles performed: 16

Generated sets of large itemsets:

Size of set of large itemsets l[1]: 11
Size of set of large itemsets l[2]: 21
Size of set of large itemsets l[3]: 6
  
```



4. Demonstration of Association rule process on dataset test.arff using apriori algorithm

Aim: This experiment illustrates some of the basic elements of association rule mining using WEKA. The sample dataset used for this example is test.arff

Step1: Open the data file in Weka Explorer. It is presumed that the required data fields have been discretized. In this example it is age attribute.

Step2: Clicking on the associate tab will bring up the interface for association rule algorithm.

Step3: We will use apriori algorithm. This is the default algorithm.

Step4: Inorder to change the parameters for the run (example support, confidence etc) we click on the text box immediately to the right of the choose button.

Dataset test.arff

@relation test

@attribute admissionyear {2005,2006,2007,2008,2009,2010}

@attribute course {cse,mech,it,ece}

@data

%

2005, cse

2005, it

2005, cse

2006, mech

2006, it

2006, ece

2007, it

2007, cse

2008, it

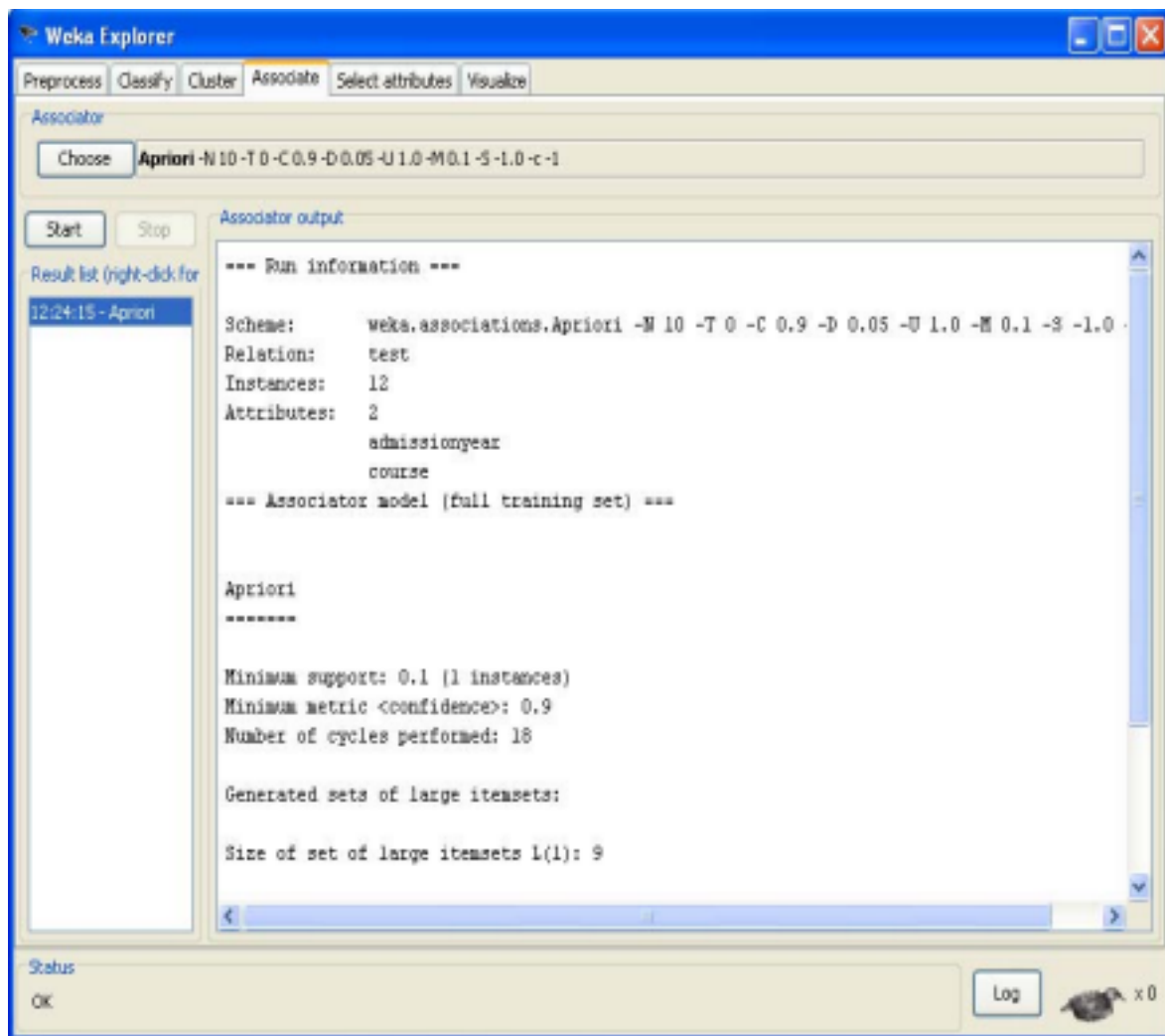
2008, cse

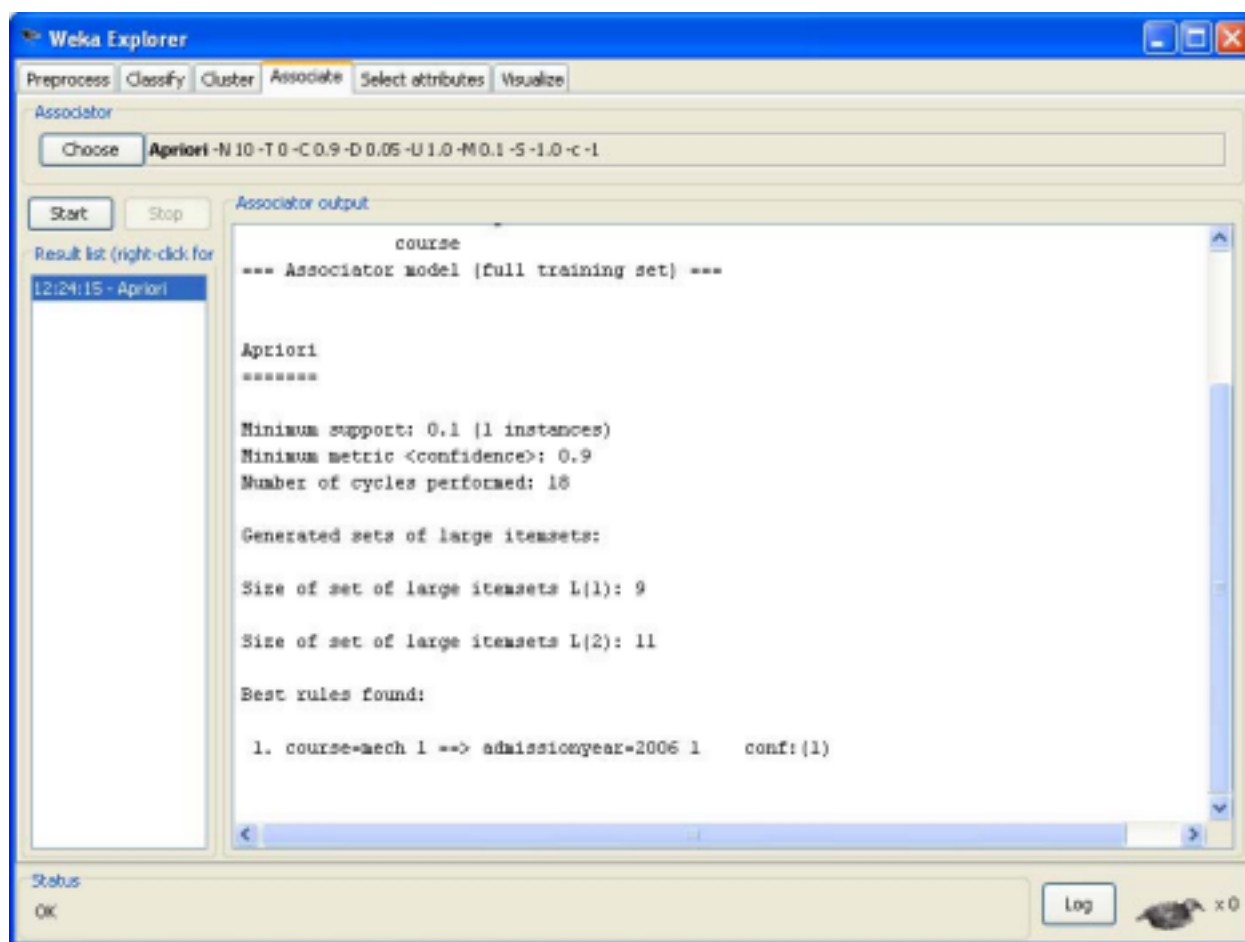
2009, it

2009, ece

%

The following screenshot shows the association rules that were generated when apriori algorithm is applied on the given dataset.





5. Demonstration of classification rule process on dataset student.arff using j48 algorithm

Aim: This experiment illustrates the use of j-48 classifier in weka. The sample data set used in this experiment is “student” data available at arff format. This document assumes that appropriate data pre processing has been performed.

Steps involved in this experiment:

Step-1: We begin the experiment by loading the data (student.arff) into weka.

Step2: Next we select the “classify” tab and click “choose” button to select the “j48” classifier.

Step3: Now we specify the various parameters. These can be specified by clicking in the text box to the right of the choose button. In this example, we accept the default values. The default version does perform some pruning but does not perform error pruning.

Step4: Under the “text” options in the main panel. We select the 10-fold cross validation as our evaluation approach. Since we don’t have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

Step-5: We now click ”start” to generate the model .the Ascii version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

Step-6: Note that the classification accuracy of model is about 69%.this indicates that we may find more work. (Either in preprocessing or in selecting current parameters for the classification)

Step-7: Now weka also lets us a view a graphical version of the classification tree. This can be done by right clicking the last result set and selecting “visualize tree” from the pop-up menu.

Step-8: We will use our model to classify the new instances.

Step-9: In the main panel under “text” options click the “supplied test set” radio button and then click the “set” button. This will pop-up a window which will allow you to open the file containing test instances.

Dataset student .arff

@relation student

@attribute age {<30,30-40,>40}

@attribute income {low, medium, high}

@attribute student {yes, no}

@attribute credit-rating {fair, excellent}

@attribute buyspc {yes, no}

@data

%

<30, high, no, fair, no

<30, high, no, excellent, no

30-40, high, no, fair, yes

>40, medium, no, fair, yes

>40, low, yes, fair, yes

>40, low, yes, excellent, no

30-40, low, yes, excellent, yes

<30, medium, no, fair, no

<30, low, yes, fair, no

>40, medium, yes, fair, yes

<30, medium, yes, excellent, yes

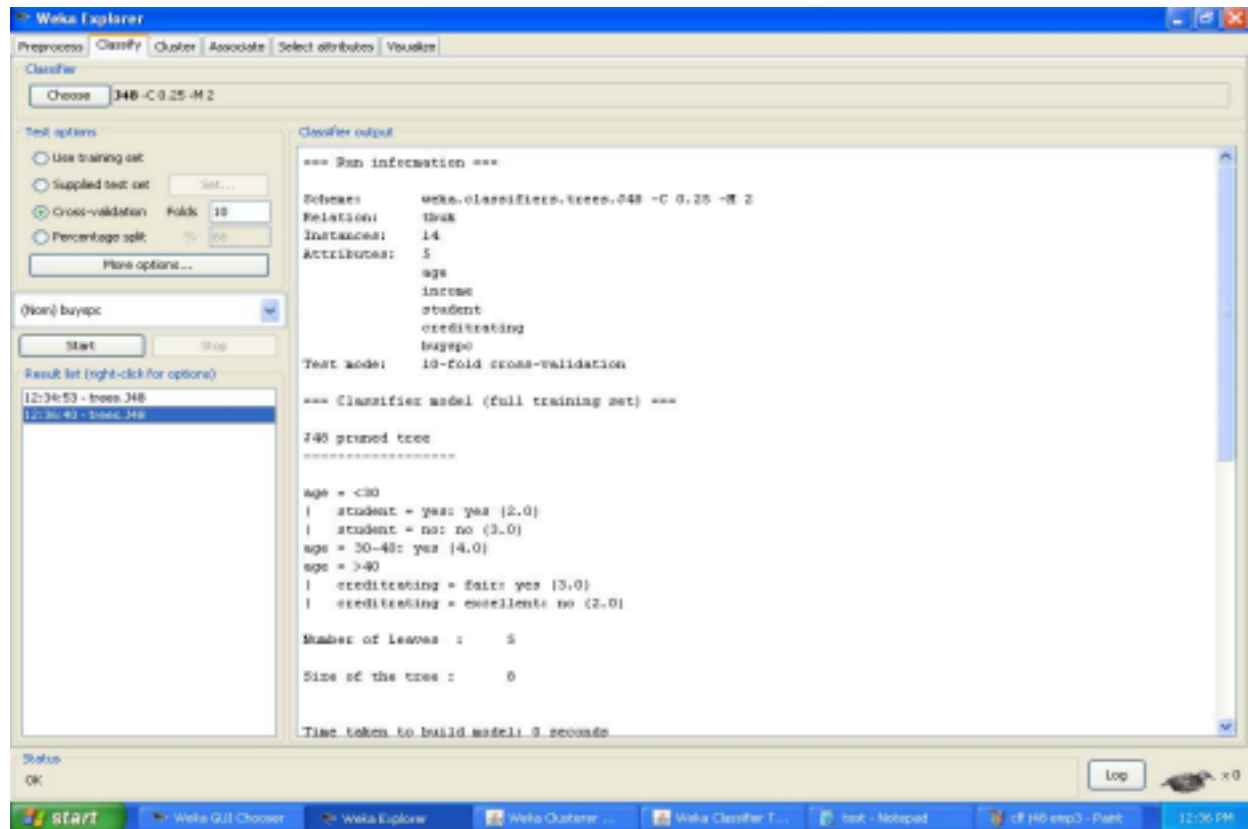
30-40, medium, no, excellent, yes

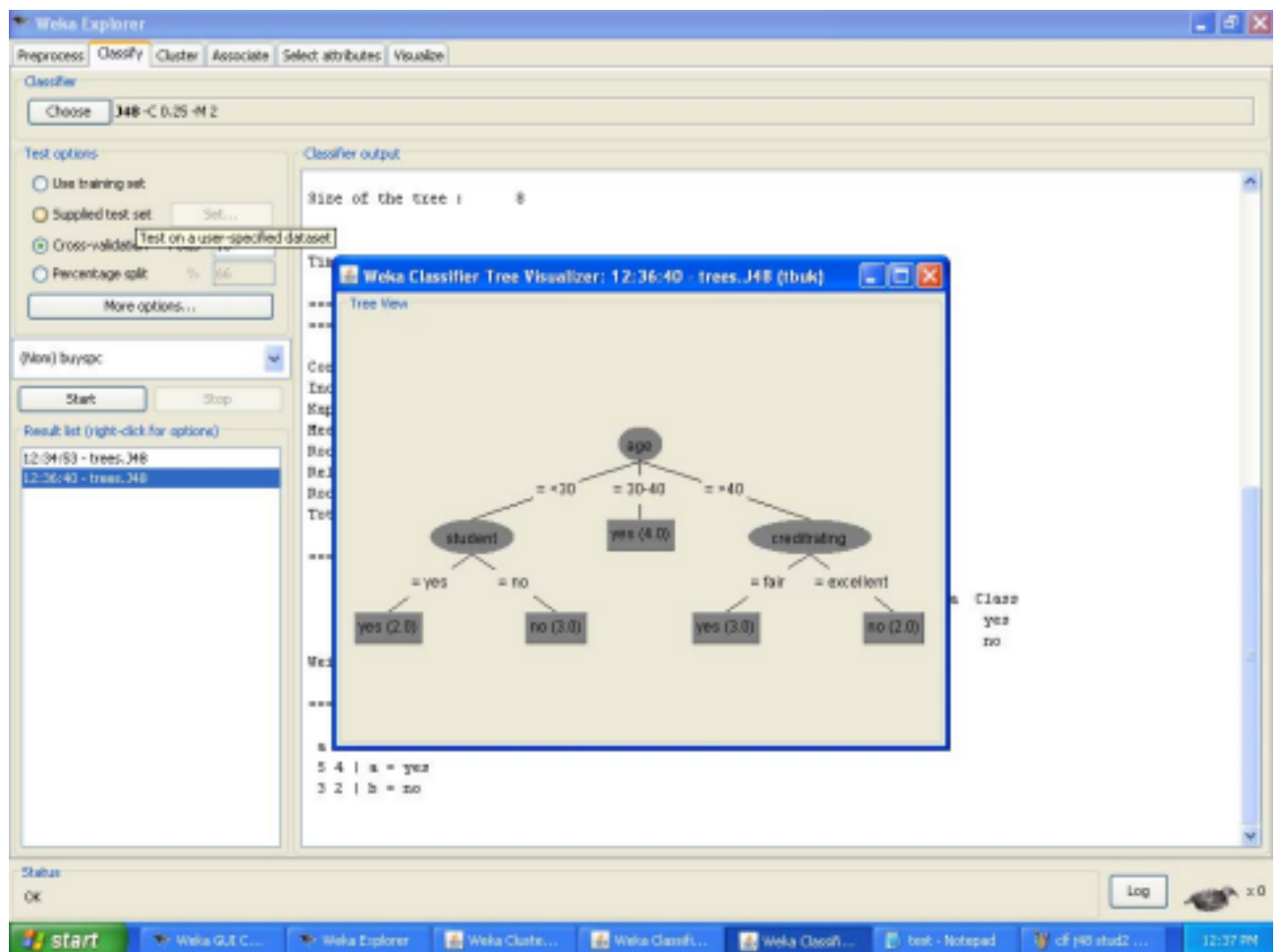
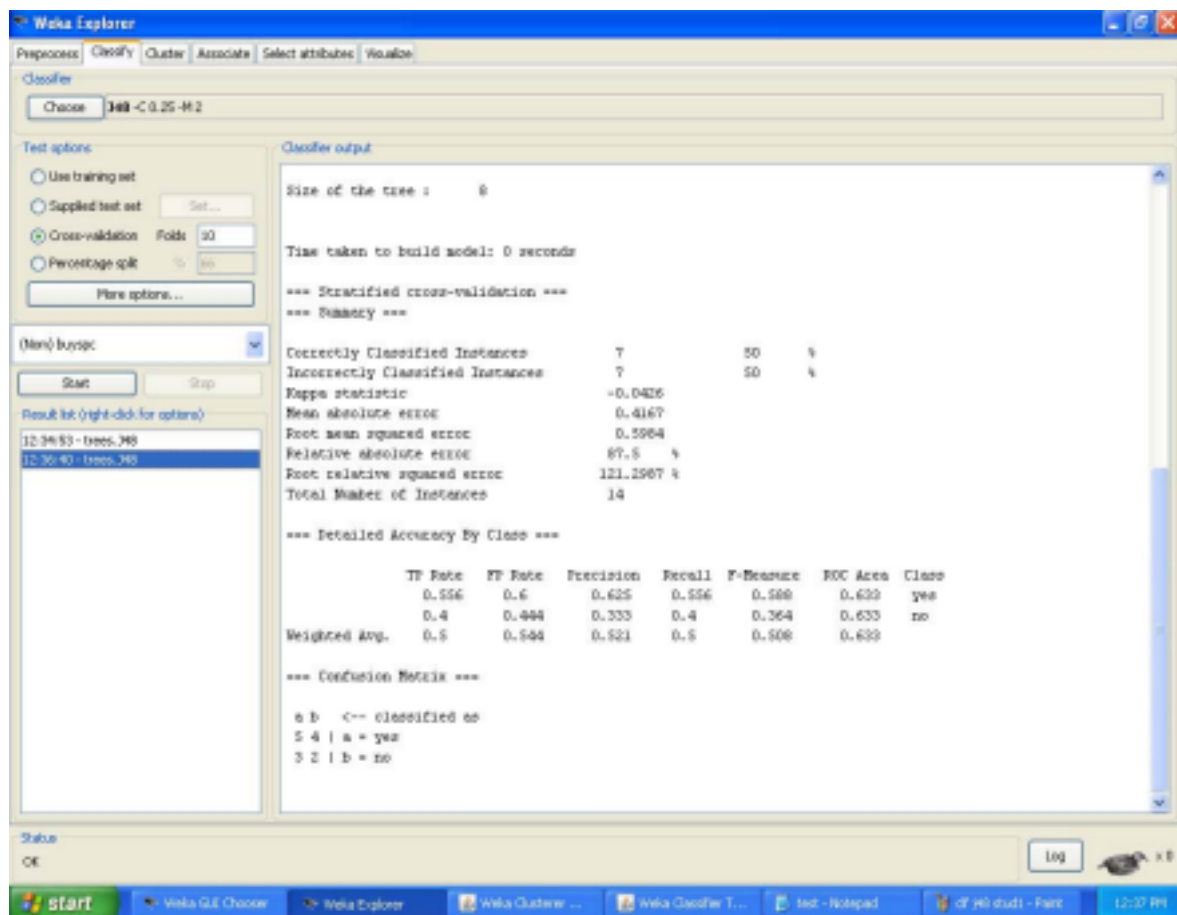
30-40, high, yes, fair, yes

>40, medium, no, excellent, no

%

The following screenshot shows the classification rules that were generated when j48 algorithm is applied on the given dataset.





6. Demonstration of classification rule process on dataset employee.arff using j48 algorithm

Aim: This experiment illustrates the use of j-48 classifier in weka. the sample data set used in this experiment is “employee” data available at arff format. This document assumes that appropriate data pre processing has been performed.

Steps involved in this experiment:

Step 1: We begin the experiment by loading the data (employee.arff) into weka.

Step2: Next we select the “classify” tab and click “choose” button to select the “j48” classifier.

Step3: Now we specify the various parameters. These can be specified by clicking in the text box to the right of the chose button. In this example, we accept the default values the default version does perform some pruning but does not perform error pruning.

Step4: Under the “text “options in the main panel. We select the 10-fold cross validation as our evaluation approach. Since we don’t have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

Step-5: We now click ”start” to generate the model .the ASCII version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

Step-6: Note that the classification accuracy of model is about 69%.this indicates that we may find more work. (Either in preprocessing or in selecting current parameters for the classification)

Step-7: Now weka also lets us a view a graphical version of the classification tree. This can be done by right clicking the last result set and selecting “visualize tree” from the pop-up menu.

Step-8: We will use our model to classify the new instances.

Step-9: In the main panel under “text “options click the “supplied test set” radio button and then click the “set” button. This will pop-up a window which will allow you to open the file containing test instances.

Data set employee.arff:

@relation employee

@attribute age {25, 27, 28, 29, 30, 35, 48}

@attribute salary {10k,15k,17k,20k,25k,30k,35k,32k}

@attribute performance {good, avg, poor} @data

%

25, 10k, poor

27, 15k, poor

27, 17k, poor

28, 17k, poor

29, 20k, avg

30, 25k, avg

29, 25k, avg

30, 20k, avg

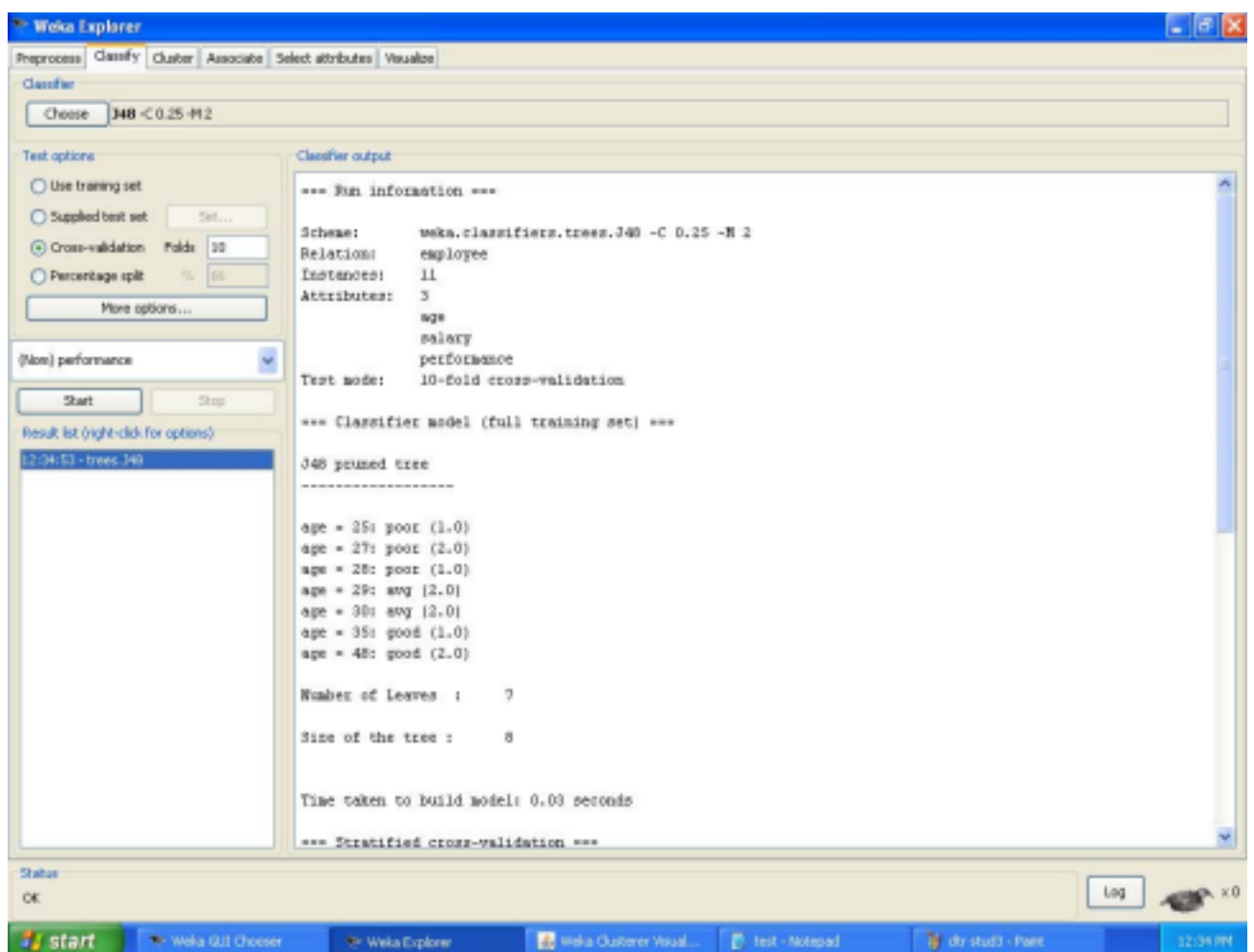
35, 32k, good

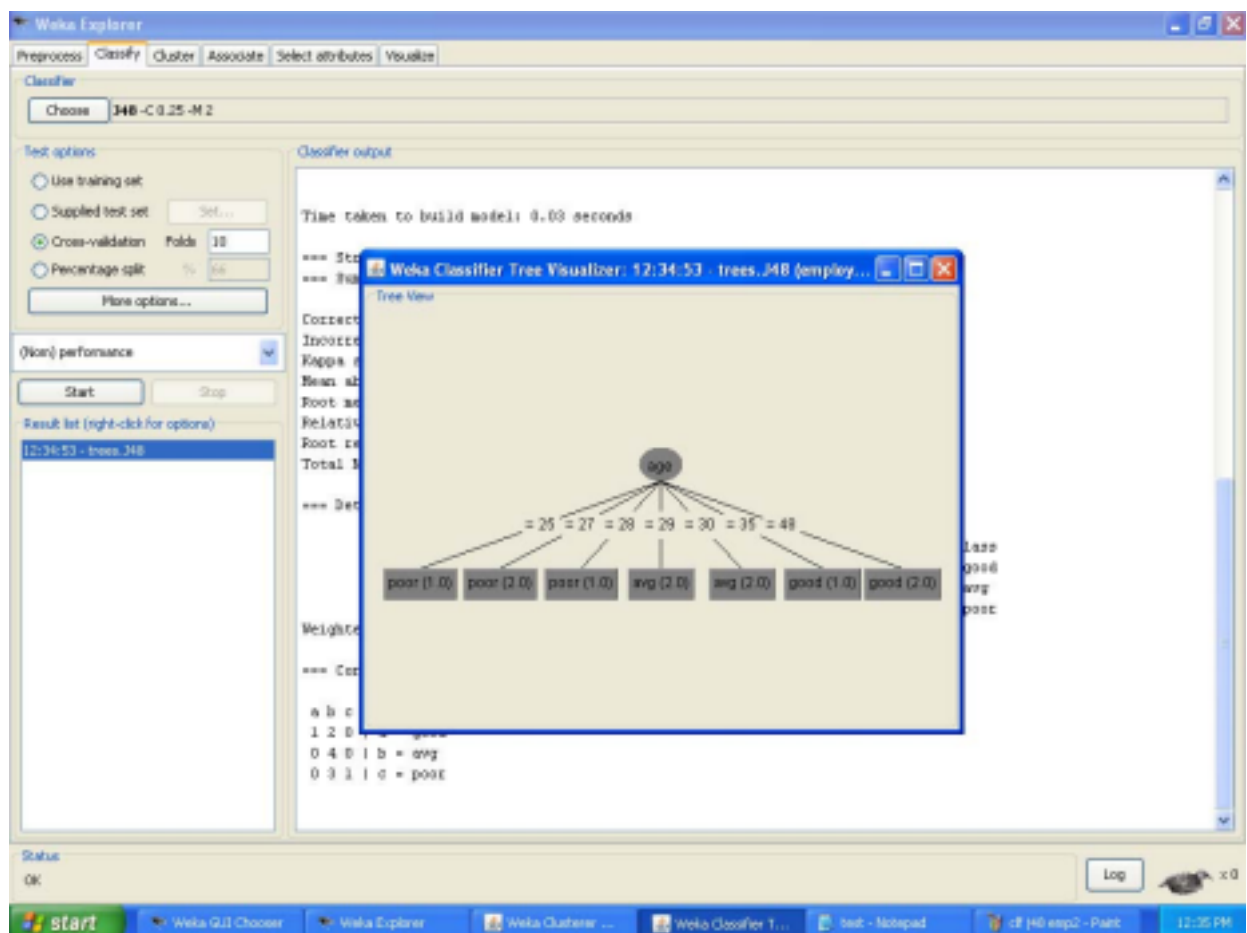
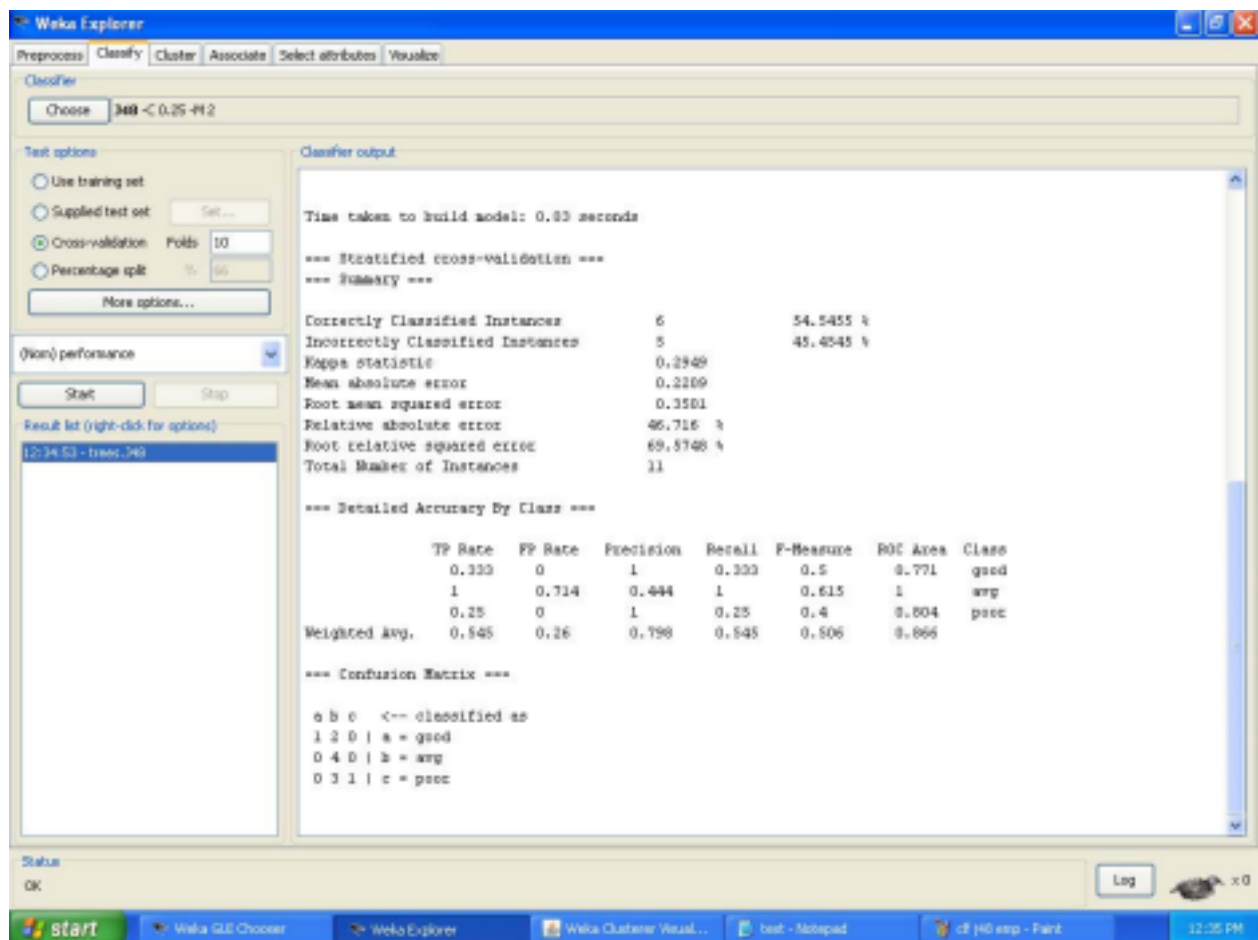
48, 34k, good

48, 32k, good

%

The following screenshot shows the classification rules that were generated when j48 algorithm is applied on the given dataset.





7. Demonstration of classification rule process on dataset employee.arff using id3 algorithm

Aim: This experiment illustrates the use of id3 classifier in weka. The sample data set used in this experiment is “employee” data available at arff format. This document assumes that appropriate data pre processing has been performed.

Steps involved in this experiment:

1. We begin the experiment by loading the data (employee.arff) into weka. Step2: next we select the “classify” tab and click “choose” button to select the “id3” classifier.

Step3: now we specify the various parameters. These can be specified by clicking in the text box to the right of the choose button. In this example, we accept the default values his default version does perform some pruning but does not perform error pruning.

Step4: under the “text “options in the main panel. We select the 10-fold cross validation as our evaluation approach. Since we don’t have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

Step-5: we now click”start”to generate the model .the ASCII version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

Step-6: note that the classification accuracy of model is about 69%.this indicates that we may find more work. (Either in preprocessing or in selecting current parameters for the classification)

Step-7: now weka also lets us a view a graphical version of the classification tree. This can be done by right clicking the last result set and selecting “visualize tree” from the pop-up menu.

Step-8: we will use our model to classify the new instances.

Step-9: In the main panel under “text “options click the “supplied test set” radio button and then click the “set” button. This will show pop-up window which will allow you to open the file containing test instances.

Data set employee.arff:

@relation employee

@attribute age {25, 27, 28, 29, 30, 35, 48}

@attribute salary {10k,15k,17k,20k,25k,30k,35k,32k}

@attribute performance {good, avg, poor} @data

%

25, 10k, poor

27, 15k, poor

27, 17k, poor

28, 17k, poor

29, 20k, avg

30, 25k, avg

29, 25k, avg

30, 20k, avg

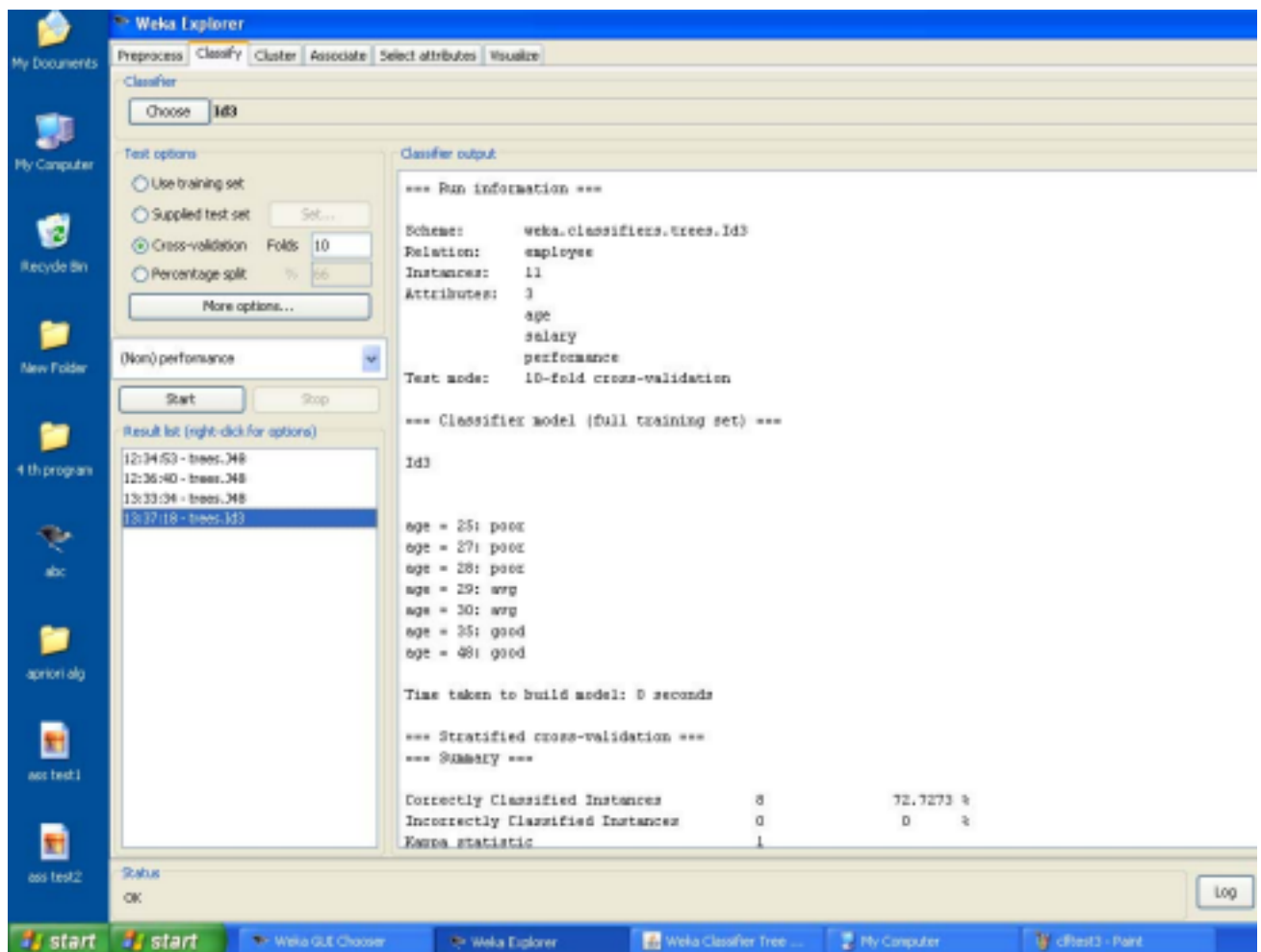
35, 32k, good

48, 34k, good

48, 32k, good

%

The following screenshot shows the classification rules that were generated when id3 algorithm is applied on the given dataset.





8.Demonstration of classification rule process on dataset employee.arff using naïve bayes algorithm

Aim: This experiment illustrates the use of naïve bayes classifier in weka. The sample data set used in this experiment is “employee”data available at arff format. This document assumes that appropriate data pre processing has been performed.

Steps involved in this experiment:

1. We begin the experiment by loading the data (employee.arff) into weka. Step2: next we select the “classify” tab and click “choose” button to select the “id3”classifier.

Step3: now we specify the various parameters. These can be specified by clicking in the text box to the right of the chose button. In this example, we accept the default values his default version does perform some pruning but does not perform error pruning.

Step4: under the “text “options in the main panel. We select the 10-fold cross validation as our evaluation approach. Since we don’t have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

Step-5: we now click”start”to generate the model .the ASCII version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

Step-6: note that the classification accuracy of model is about 69%.this indicates that we may find more work. (Either in preprocessing or in selecting current parameters for the classification)

Step-7: now weka also lets us a view a graphical version of the classification tree. This can be done by right clicking the last result set and selecting “visualize tree” from the pop-up menu.

Step-8: we will use our model to classify the new instances.

Step-9: In the main panel under “text “options click the “supplied test set” radio button and then click the “set” button. This will show pop-up window which will allow you to open the file containing test instances.

Data set employee.arff:

@relation employee

@attribute age {25, 27, 28, 29, 30, 35, 48}

@attribute salary {10k,15k,17k,20k,25k,30k,35k,32k}

@attribute performance {good, avg, poor} @data

%

25, 10k, poor

27, 15k, poor

27, 17k, poor

28, 17k, poor

29, 20k, avg

30, 25k, avg

29, 25k, avg

30, 20k, avg

35, 32k, good

48, 34k, good

48, 32k, good

%

The following screenshot shows the classification rules that were generated when naive bayes algorithm is applied on the given dataset.



9. Demonstration of clustering rule process on dataset iris.arff using simple k-means

Aim: This experiment illustrates the use of simple k-mean clustering with Weka explorer. The sample data set used for this example is based on the iris data available in ARFF format. This document assumes that appropriate preprocessing has been performed. This iris dataset

includes 150 instances.

Steps involved in this Experiment

Step 1: Run the Weka explorer and load the data file iris.arff in preprocessing interface.

Step 2: In order to perform clustering select the 'cluster' tab in the explorer and click on the choose button. This step results in a dropdown list of available clustering algorithms.

Step 3 : In this case we select 'simple k-means'.

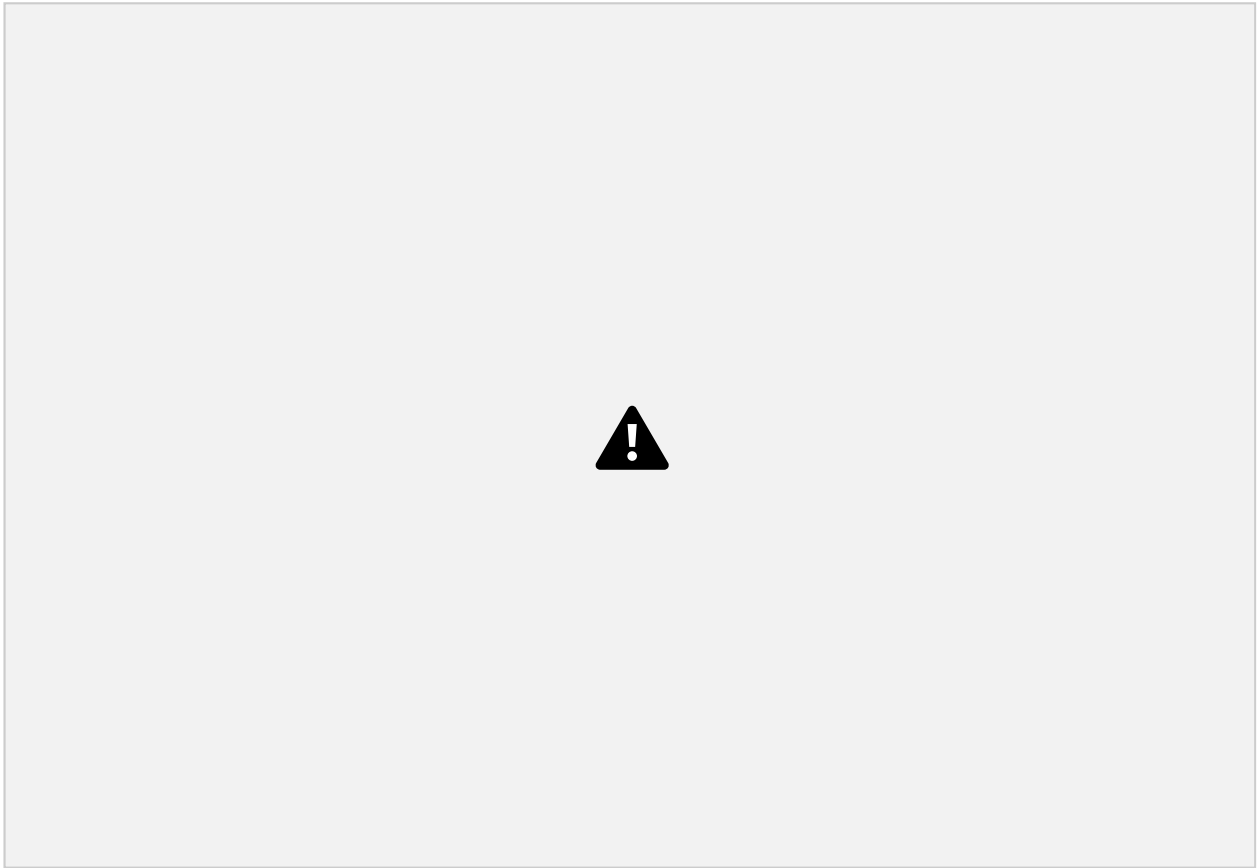
Step 4: Next click in text button to the right of the choose button to get popup window shown in the screenshots. In this window we enter six on the number of clusters and we leave the value of the seed on as it is. The seed value is used in generating a random number which is used for making the internal assignments of instances of clusters.

Step 5 : Once of the option have been specified. We run the clustering algorithm there we must make sure that they are in the 'cluster mode' panel. The use of training set option is selected and then we click 'start' button. This process and resulting window are shown in the following screenshots.

Step 6 : The result window shows the centroid of each cluster as well as statistics on the number and the percent of instances assigned to different clusters. Here clusters centroid are means vectors for each clusters. This clusters can be used to characterized the cluster. For eg, the centroid of cluster1 shows the class iris.versicolor mean value of the sepal length is 5.4706, sepal width 2.4765, petal width 1.1294, petal length 3.7941.

Step 7: Another way of understanding characteristics of each cluster through visualization ,we can do this, try right clicking the result set on the result. List panel and selecting the visualize cluster assignments.

The following screenshot shows the clustering rules that were generated when simple k means algorithm is applied on the given dataset.



Interpretation of the above visualization

From the above visualization, we can understand the distribution of sepal length and petal length in each cluster. For instance, for each cluster is dominated by petal length. In this case by changing the color dimension to other attributes we can see their distribution within each of the cluster.

Step 8: We can assure that resulting dataset which included each instance along with its assigned cluster. To do so we click the save button in the visualization window and save the result iris k-mean. The top portion of this file is shown in the following figure.



10. Demonstration of clustering rule process on dataset student.arff using simple k means

Aim: This experiment illustrates the use of simple k-mean clustering with Weka explorer. The sample data set used for this example is based on the student data available in ARFF format. This document assumes that appropriate preprocessing has been performed. This student dataset includes 14 instances.

Steps involved in this Experiment

Step 1: Run the Weka explorer and load the data file student.arff in preprocessing interface.

Step 2: In order to perform clustering select the 'cluster' tab in the explorer and click on the choose button. This step results in a dropdown list of available clustering algorithms.

Step 3 : In this case we select 'simple k-means'.

Step 4: Next click in text button to the right of the choose button to get popup window shown in the screenshots. In this window we enter six on the number of clusters and we leave the value of the seed on as it is. The seed value is used in generating a random number which is used for making the internal assignments of instances of clusters.

Step 5 : Once of the option have been specified. We run the clustering algorithm there we must make sure that they are in the 'cluster mode' panel. The use of training set option is selected and then we click 'start' button. This process and resulting window are shown in the

following screenshots.

Step 6 : The result window shows the centroid of each cluster as well as statistics on the number and the percent of instances assigned to different clusters. Here clusters centroid are means vectors for each clusters. This clusters can be used to characterized the cluster.

Step 7: Another way of understanding characteristics of each cluster through visualization ,we can do this, try right clicking the result set on the result. List panel and selecting the visualize cluster assignments.

Interpretation of the above visualization

From the above visualization, we can understand the distribution of age and instance number in each cluster. For instance, for each cluster is dominated by age. In this case by changing the color dimension to other attributes we can see their distribution with in each of the cluster.

Step 8: We can assure that resulting dataset which included each instance along with its assign cluster. To do so we click the save button in the visualization window and save the result student k-mean .The top portion of this file is shown in the following figure.

Dataset student .arff

@relation student

@attribute age {<30,30-40,>40}

@attribute income {low,medium,high}

@attribute student {yes,no}

@attribute credit-rating {fair,excellent}

@attribute buyspc {yes,no}

@data

%

<30, high, no, fair, no

<30, high, no, excellent, no

30-40, high, no, fair, yes

>40, medium, no, fair, yes

>40, low, yes, fair, yes

>40, low, yes, excellent, no

30-40, low, yes, excellent, yes

<30, medium, no, fair, no

<30, low, yes, fair, no

>40, medium, yes, fair, yes

<30, medium, yes, excellent, yes

30-40, medium, no, excellent, yes

30-40, high, yes, fair, yes

>40, medium, no, excellent, no

%

The following screenshot shows the clustering rules that were generated when simple k means algorithm is applied on the given dataset.

