ADIGRAT UNIVERSITY

DATAMINING USING WEKA LAB MANUAL

Introduction

WEKA is open source java code created by researchers at the University of Waikato in New Zealand. It provides many different machine learning algorithms, including the following classifiers:

- Decision tree (j4.8, an extension of C4.5)
- MLP, aka multiple layer perceptron (a type of neural net)
- Naïve bayes
- Rule induction algorithms such as JRip
- Support vector machine
- And many more

The GUI WEKA

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Weka's main GUI applications and supporting tools. If one prefers a MDI ("multiple document inter face") appearance, then this is provided by an alternative launcher called "Main" (class weka.gui.Main). The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.

The buttons can be used to start the following applications:

• ***Explorer*** ***:*** An environment for exploring data with WEKA (the rest of this documentation deals with this application in more detail).

• ***Experimenter***: An environment for performing experiments and conducting statistical tests between learning schemes.

• ***KnowledgeFlow:*** This environment supports essentially the same functions as the Explorer but with a drag-and-drop interface. One advantage is that it supports incremental learning.

• ***SimpleCLI:*** Provides a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface. The menu consists of four sections:

WEKA Explorer

The user interface

Section Tabs

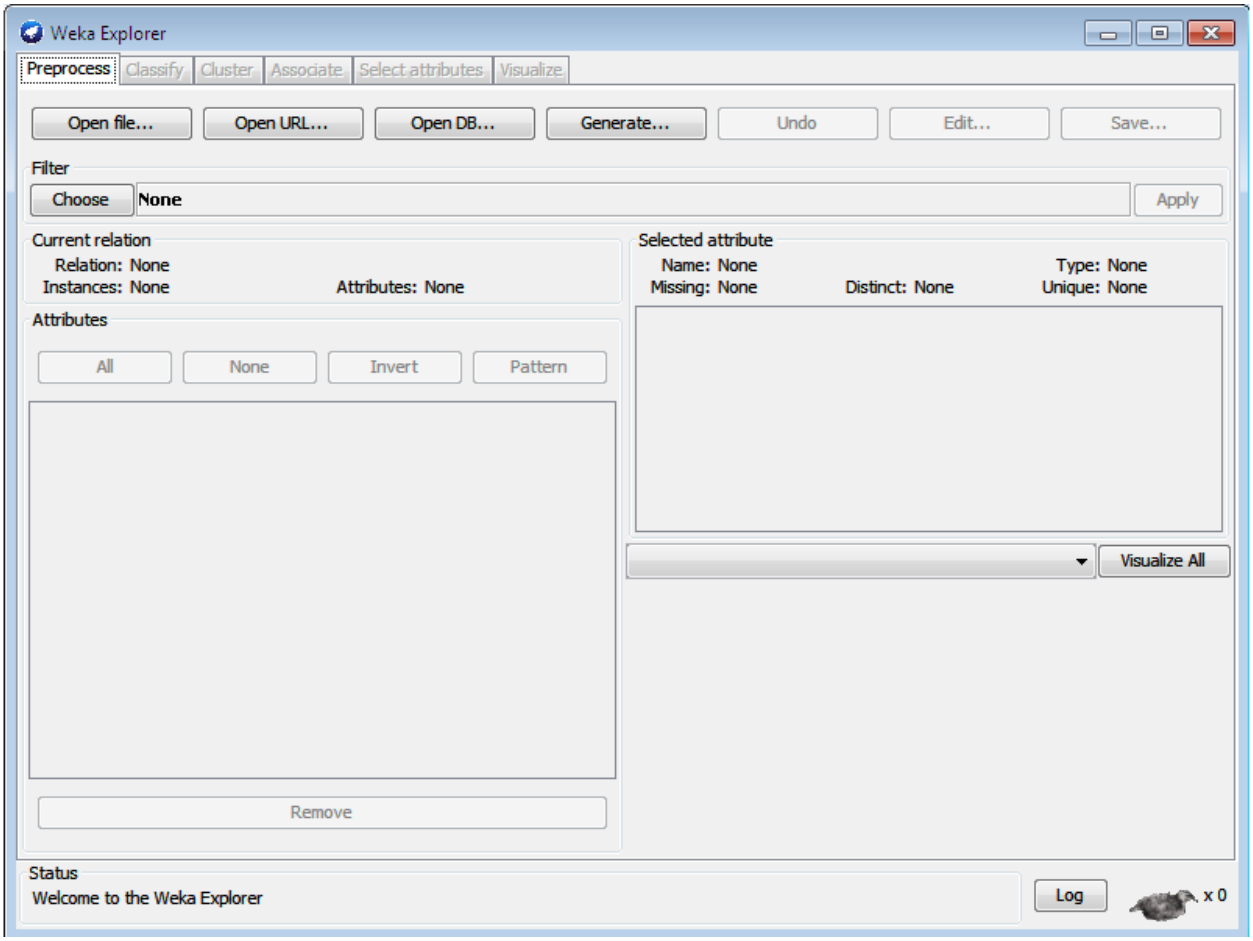
At the very top of the window, just below the title bar, is a row of tabs. When the Explorer is first started only the first tab is active; the others are grayed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.

The tabs are as follows:

1. ***Preprocess.*** Choose and modify the data being acted on.
2. ***Classify.*** Train and test learning schemes that classify or perform regression.
3. ***Cluster.*** Learn clusters for the data.
4. ***Associate.*** Learn association rules for the data.
5. ***Select attributes*****.** Select the most relevant attributes in the data.
6. ***Visualize.*** View an interactive 2D plot of the data.

Once the tabs are active, clicking on them flicks between different screens, on which the respective actions can be performed. The bottom area of the window (including the status box, the log button, and the Weka bird) stays visible regardless of which section you are in. The Explorer can be easily extended with custom tabs.

Preprocessing



Loading Data

The first four buttons at the top of the preprocess section enable you to load data into WEKA

1. ***Open file***: Brings up a dialog box allowing you to browse for the data file on the local file system.
2. ***Open URL***: Asks for a uniform resource locator address for where the data is stored.
3. ***Open DB***: Reads data from a database.(Note that to make this work you might have to edit the file in weka/experiment/databseutils.props.)
4. ***Generate***: Enable you to generate artificial data from a variety of Data generators. Using the Open file…button you can read files in a variety of formats:WEKA's ARFF format, CSV format, C4.5 format, or serialized instances format. ARFF files typically have a .arff extension, CSV files a.csv extension, C4.5 files a.data and .names extension and serialized instances objects a. bsi extension.

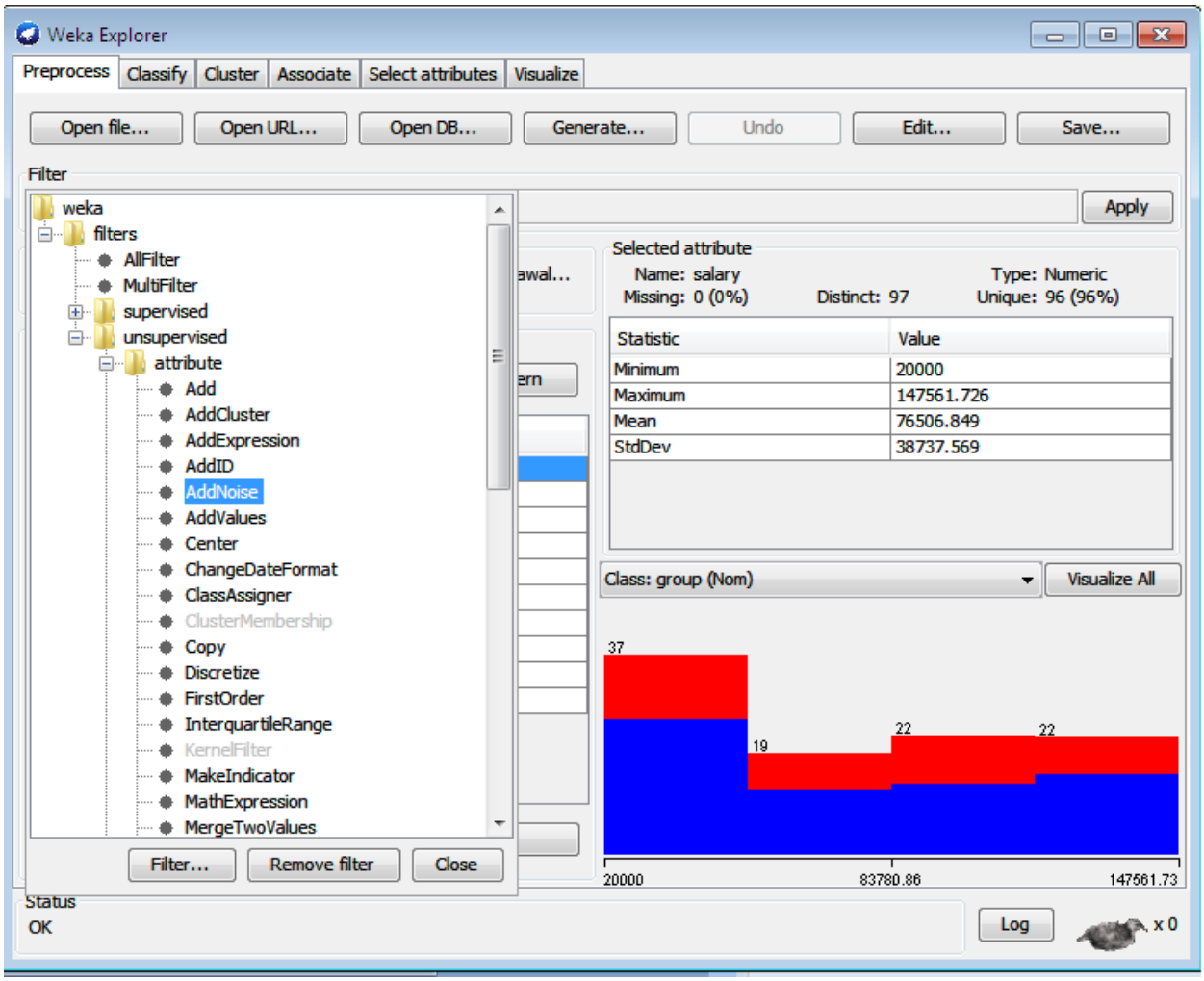
Preprocessing

1. All. All boxes are ticked.

2. None. All boxes are cleared (unticked).

3. Invert. Boxes that are ticked become unticked and vice versa.

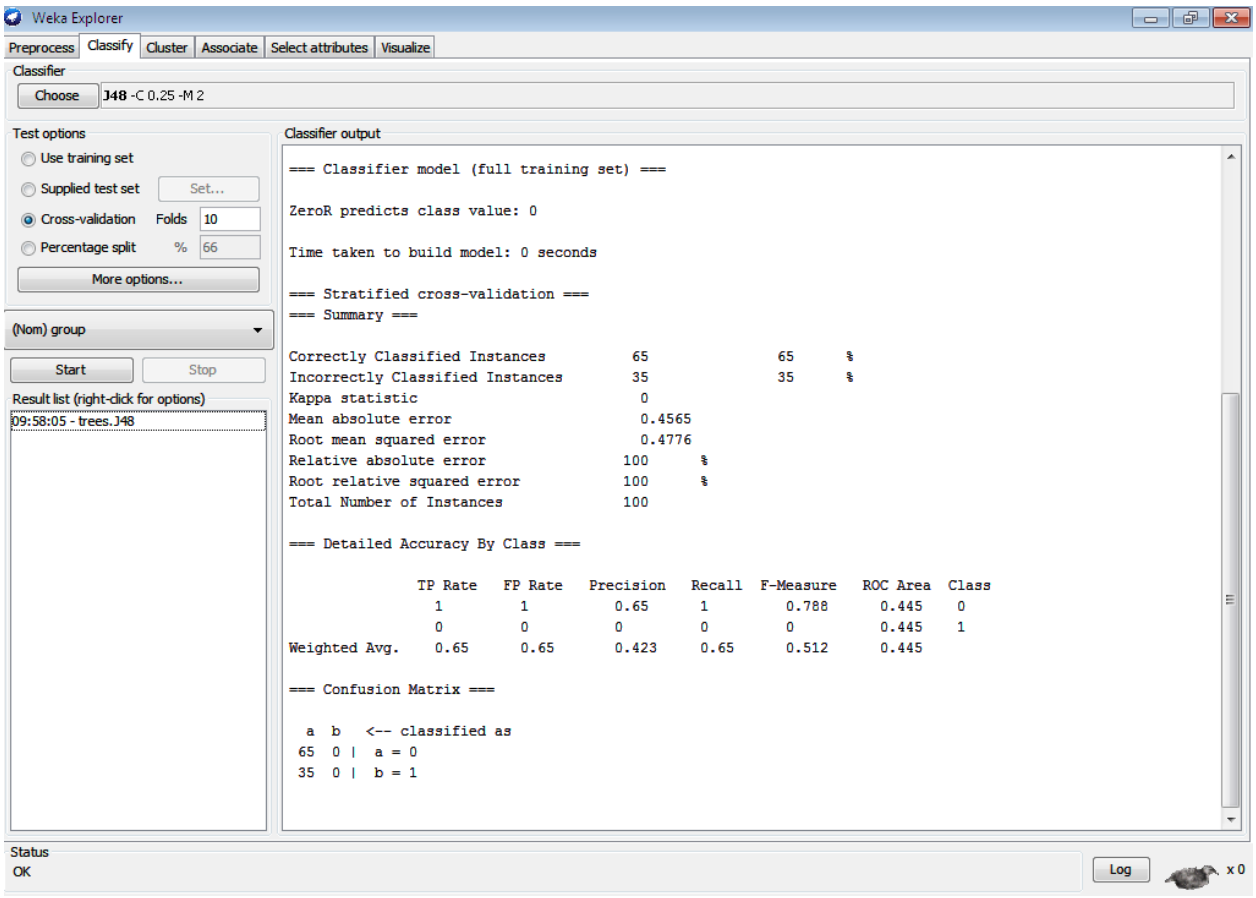
4. Pattern. Enables the user to select attributes based on a Perl 5 Regular Expression. E.g:- .* id selects all attributes which name ends with id. Once the desired attributes have been selected, they can be removed by clicking the Remove button below the list of attributes. Note that this can be undone by clicking the Undo button, which is located next to the Edit button in the top-right corner of the Preprocess panel.



The preprocess section allows filters to be defined that transform the data in various ways. The Filter box is used to set up the filters that are required. At the left of the Filter box is a Choose button. By clicking this button it is possible to select one of the filters in WEKA. Once a filter has been selected, its name and options are shown in the field next to the Choose button. Clicking on this box with the left mouse button brings up a GenericObjectEditor dialog box. A click with the right mouse button (or

Alt+Shift+left click) brings up a menu where you can choose, either to display the properties in a GenericObjectEditor dialog box, or to copy the current setup string to the clipboard

Classification



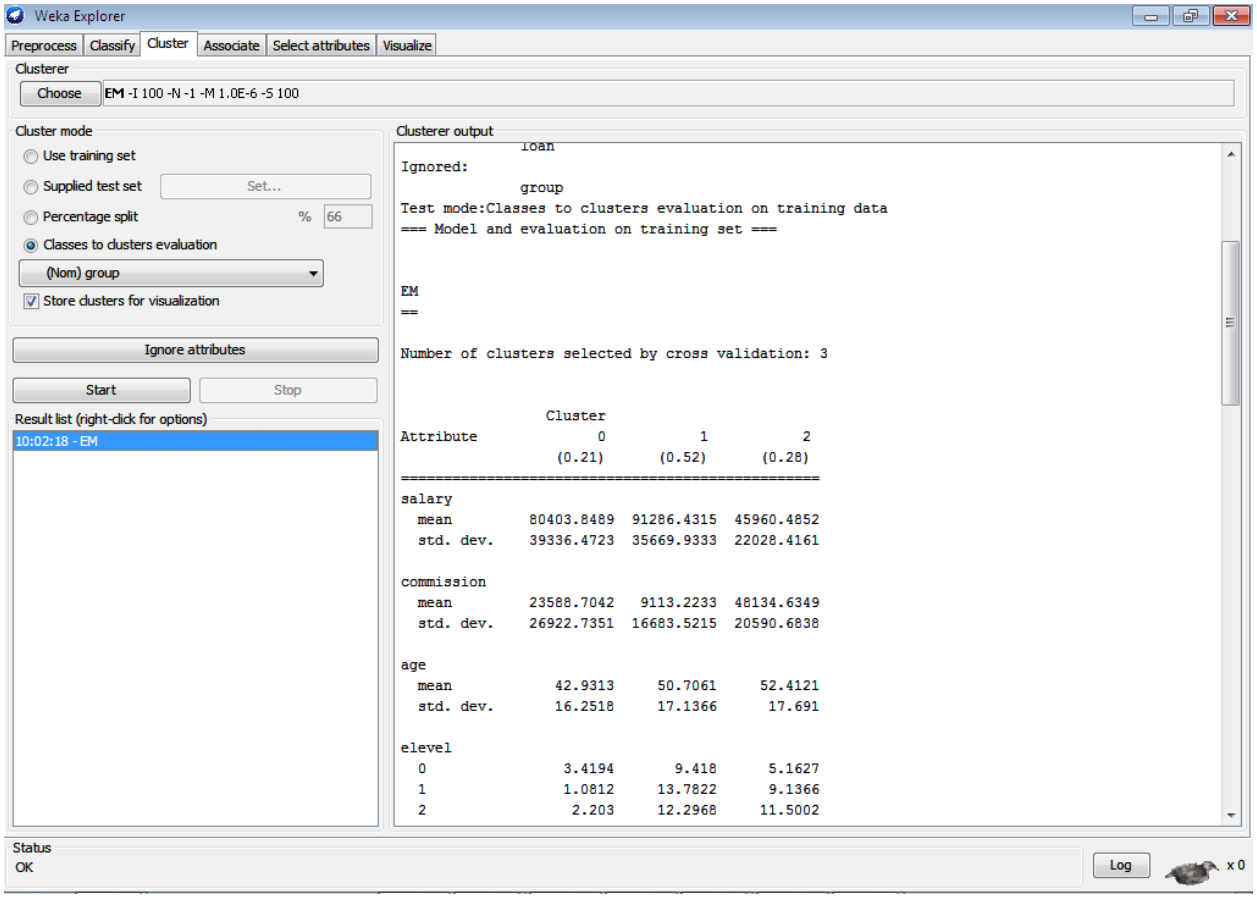
Selecting a Classifier

At the top of the classify section is the Classifier box. This box has a text field that gives the name of the currently selected classifier, and its options. Clicking on the text box with the left mouse button brings up a GenericObjectEditor dialog box, just the same as for filters that you can use to configure the options of the current classifier. With a right click (or Alt+Shift+left click) you can once again copy the setup string to the clipboard or display the properties in a GenericObjectEditor dialog box. The Choose button allows you to choose one of the classifiers that are available in WEKA.

Clustering

Selecting a Clusterer

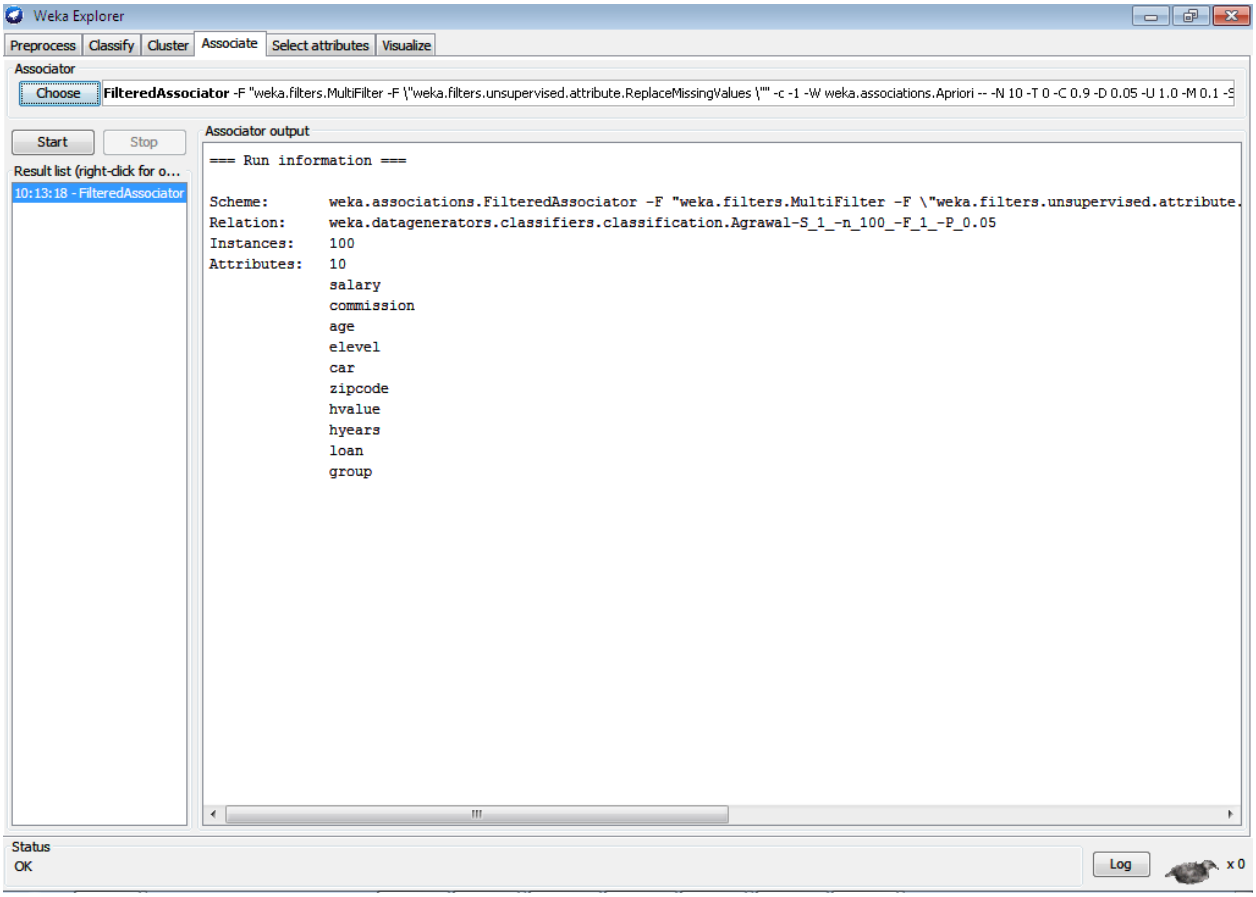
By now you will be familiar with the process of selecting and configuring objects. Clicking on the clustering scheme listed in the Clusterer box at the top of the window brings up a GenericObjectEditor dialog with which to choose a new clustering scheme



Cluster Modes

The cluster mode box is used to choose what to cluster and how to evaluate the results. The three options are the same as for classification: use training set, supplied test set and percentage split except that now the data is assigned to clusters instead of trying to predict a specific class. The fourth mode, classes to clusters evaluation, compares how well the chosen clusters match up with a pre-assigned class in the data. The drop-down box below this option selects the class just as in the classify panel. An additional option in the cluster mode box, the store clusters for visualization tick box, determines whether or not it will be possible to visualize the clusters once training is complete. When dealing with datasets that are so large that memory becomes a problem it may be helpful to disable this option.

Associating



Setting Up

This panel contains schemes for learning association rules, and the learners are chosen and configured in the same way as the clusterers, filters, and classifiers in the other panels.

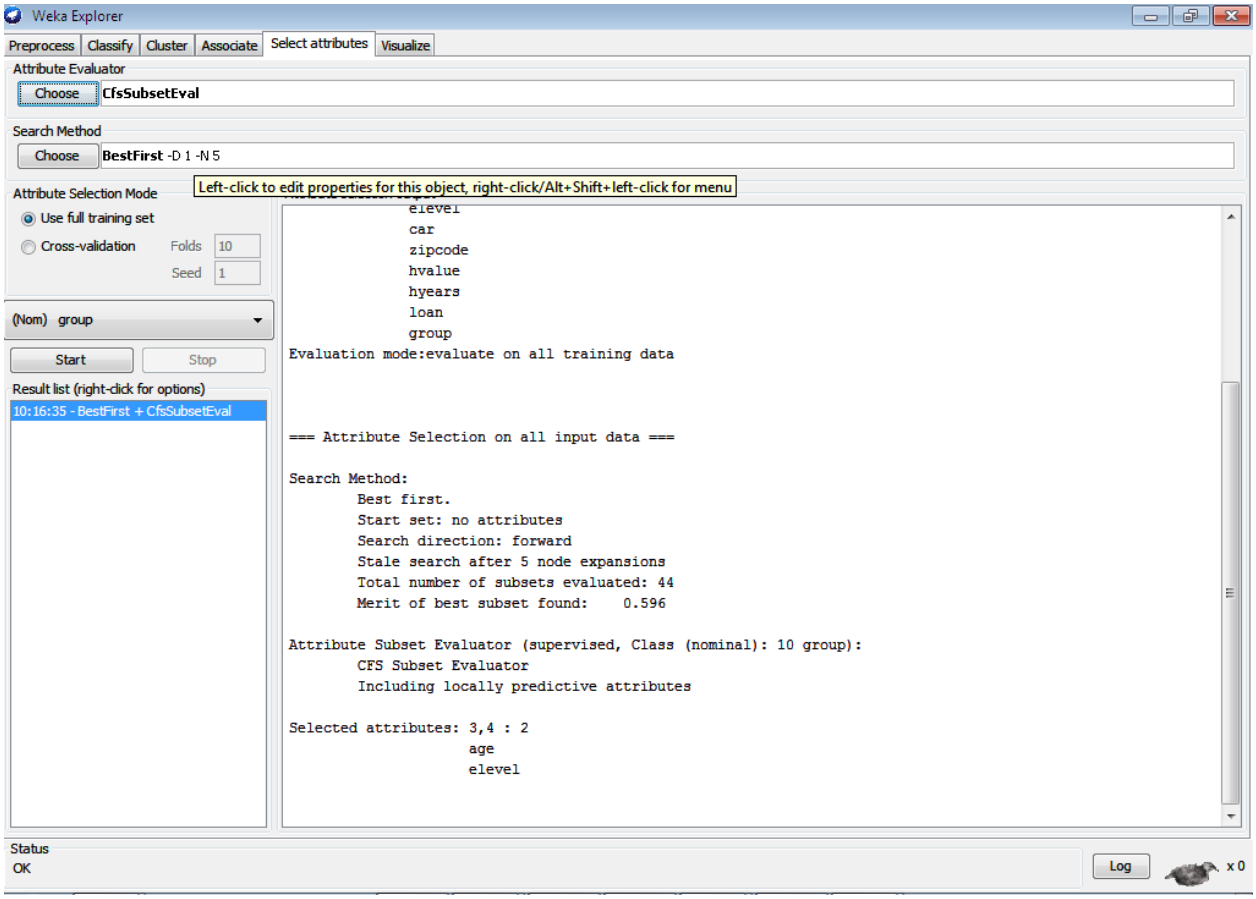
Learning Associations

Once appropriate parameters for the association rule learner have been set, click the Start button. When complete, right-clicking on an entry in the result list allows the results to be viewed or saved.

Selecting Attributes

Searching and Evaluating

Attribute selection involves searching through all possible combinations of attributes in the data to find which subset of attributes works best for prediction. To do this, two objects must be set up: an attribute evaluator and a search method. The evaluator determines what method is used to assign a worth to each subset of attributes. The search method determines what style of search is performed.

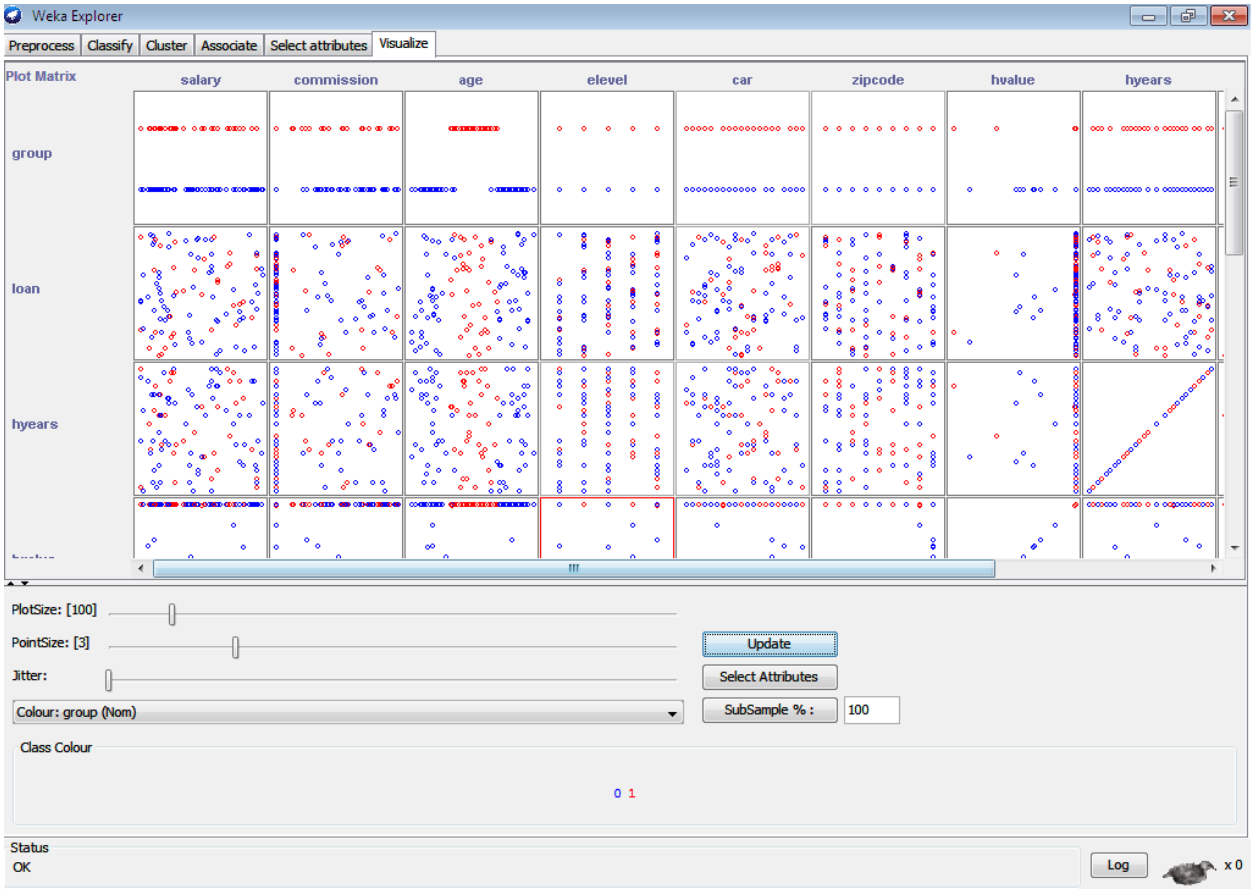


Options

The Attributes selection mode box has two options

1. ***Use full training set :*** The worth of the attribute subset is determined using the full set of training data
2. ***Cross validation:*** The worth of the attribute subset is determined by a process of cross-validation. The fold and seed fields set the number of folds to use and the random seed used when shuffling the data. As with classify, there is a drop-down box that can be used to specify which attribute to treat as the class.

Visualizing



WE K !'s vi sual ization section allows you to visualize 2D plots of the current relation.

The scatter plot matrix

When you select the Visualize panel, it shows a scatter plot matrix for all the attributes, colour coded according to the currently selected class. It is possible to change the size of each individual 2D plot and the point size, and to randomly jitter the data (to uncover obscured points). It also possible to change the attribute used to colour the plots, to select only a subset of attributes for inclusion in the scatter plot matrix, and to sub sample the data. Note that changes will only come into effect once the Update button has been pressed.

Selecting an individual 2D scatter plot

When you click on a cell in the scatter plot matrix, this will bring up a separate window with a visualization of the scatter plot you selected. (We described above how to visualize particular results in a separate window—for example, classifier errors—the same visualization controls are used here.) Data points are plotted in the main area of the window. At the top are two drop-down list buttons for selecting the axes to plot. The one on the left shows which attribute is used for the x-axis; the one on the right shows which is used for the y-axis. Beneath the x-axis selector is a drop-down list for choosing the colour scheme. This allows you to colour the points based on the attribute selected. Below the plot area, a legend describes what values the colours correspond to. If the values are discrete, you can modify the colour used for each one by clicking on them and making an appropriate selection in the window that pops up. To the right of the plot area is a series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution of values of the attribute. These values are randomly scattered vertically to help you see concentrations of points. You can choose what axes are used in the main graph by clicking on these strips. Left-clicking an attribute strip changes the x-axis to that attribute, whereas right-clicking changes the y-axis. The 'X ' and 'Y ' written beside the strips shows what the current axes are ('B ' i s used f or 'both X and Y') . Above the attribute strips is a slider labelled Jitter, which a random displacement is given to all points in the plot. Dragging it to the right increases the amount of jitter, which is useful for spotting concentrations of points? Without jitter, a million instances at the same point would look no different to just a single lonely instance.

Experimenter

Introduction

The Weka Experiment Environment enables the user to create, run, modify, and analyze experiments in a more convenient manner than is possible when processing the schemes individually. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyze the results to determine if one of the schemes is (statistically) better than the other schemes. The Experimenter comes in two flavors, either with a simple interface that provides most of the functionality one needs for experiments, or with an interface with full access to the Experimenter 'scalabilities. You can choose between those two with the Experiment Configuration Mode radio buttons:

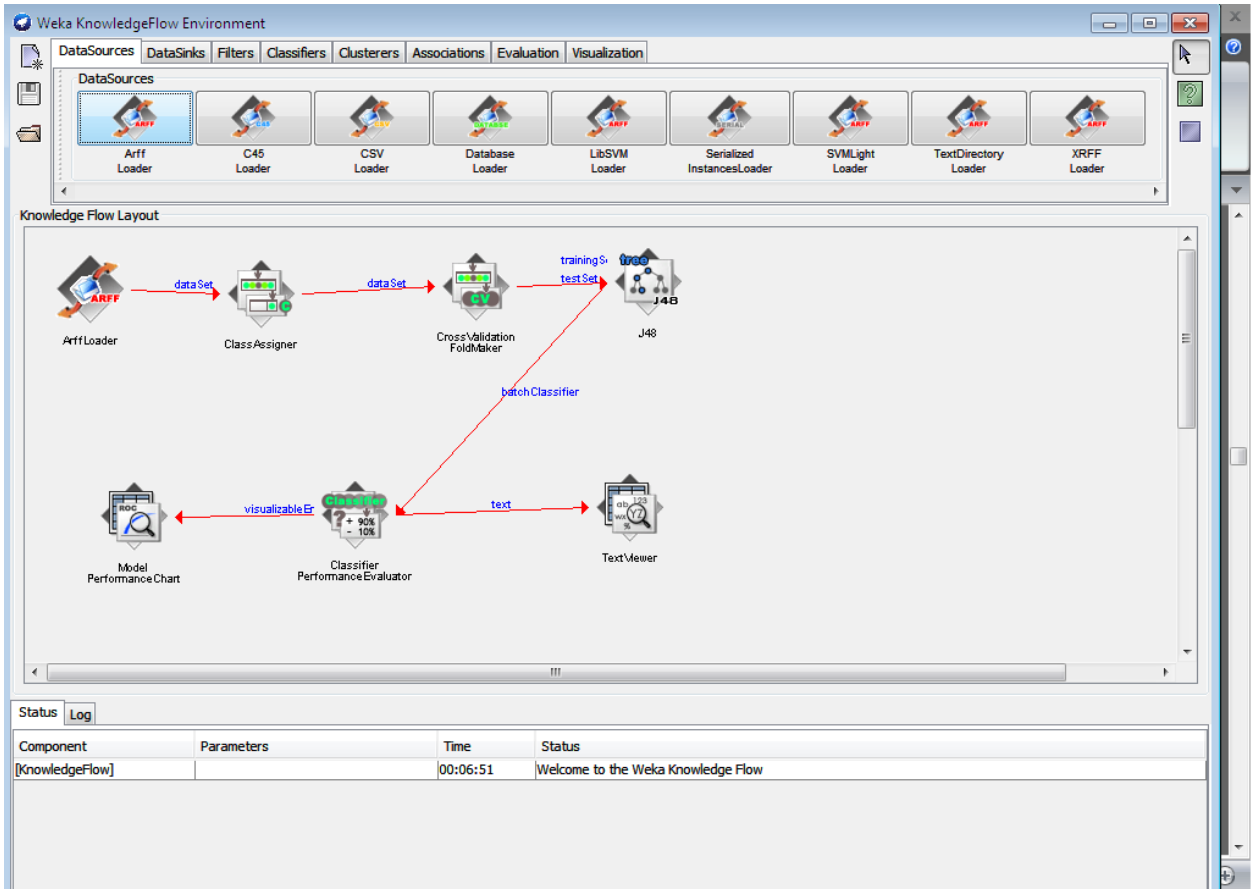
- ***Simple***
- ***Advanced***

Both setups allow you to setup standard experiments that are run locally on a single machine, or remote experiments, which are distributed between several hosts. The distribution of experiments cuts down the time the experiments will take until completion, but on the other hand the setup takes more time.

KnowledgeFlow

Introduction

The KnowledgeFlow provides an alternative to the Explorer as a graphical front end to WEKA is core algorithms. The KnowledgeFlow presents a data-flow inspired interface to WEKA. The user can select WEKA components from a palette place them on a layout canvas and connect them together in order to form a knowledge flow for processing and analyzing data. It present, all of WEKA's classifiers, filters, clusters, associates, loader s and savers are available in the KnowledgeFlow along with some extra tools.



Features

The KnowledgeFlow offers the following features:

• Intuitive data flow style layout

• Process data in batches or incrementally

• process multiple batches or streams in parallel (each separate flow executes in its own thread)

• process multiple streams sequentially via a user-specified order of execution

• Chain filters together

• View models produced by classifiers for each fold in a cross validation

• visualize performance of incremental classifiers during processing (scrolling plots of classification accuracy, RMS error, predictions etc.)

• Plugin “ perspectives” that add major new functionality (e.g . 3D data visualization, time series forecasting environment etc.)

ArffViewer

The ArffViewer is a little tool for viewing ARFF files in a tabular format. The advantage of this kind of display over the file representation is, that attribute name, type and data are directly associated in columns and not separated in definition and data part. But the viewer is not only limited to viewing multiple files at once, but also provides simple editing functionality, like sorting and deleting.

[Pending]

Description of the German credit dataset.

Title: German Credit data

Source Information: Professor Dr. Hans Hofmann, Institute flur Statistikund "Okonometrie University" at Hamburg FB Wirtschaftswissenschaften, Von-Melle-Park 5 2000 Hamburg 13

Number of Instances: 1000

Number of Attributes german: 20 (7 numerical, 13 categorical) Number of Attributes german.numer: 24 (24 numerical)

Attribute description for german

Attribute 1: (qualitative) Status of existing checking account

A11 : ... < 0 DM A12 : 0 <= ... < 200 DM

A13 : ... >= 200 DM / salary assignments for at least 1 year A14 : no checking account

Attribute 2: (numerical) Duration in month

Attribute 3: (qualitative) Credit history

A30 : no credits taken/ all credits paid back duly A31 : all credits at this bank paid back duly

A32 : existing credits paid back duly till now A33 : delay in paying off in the past

A34 : critical account/ other credits existing (not at this bank)

Attribute 4: (qualitative) Purpose

A40 : car (new) A41 : car (used) A42 : furniture/equipment A43 : radio/television

A44 : domestic appliances A45 : repairs A46 : education A47 : (vacation - does not exist?)

A48 : retraining A49 : business A410 : others

Attribute 5: (numerical) Credit amount

Attribute 6: (qualitative) Savings account/bonds

A61 : ... < 100 DM A62 : 100 <= ... < 500 DM A63 : 500 <= ... < 1000 DM

A64 : .. >= 1000 DM A65 : unknown/ no savings account

Attribute 7: (qualitative) Present employment since

A71 : unemployed A72 : ... < 1 year A73 : 1 <= ... < 4 years

A74 : 4 <= ... < 7 years A75 : .. >= 7 years

Attribute 8: (numerical) Installment rate in percentage of disposable income

Attribute 9: (qualitative) Personal status and sex

A91 : male : divorced/separated A92 : female : divorced/separated/married

A93 : male : single A94 : male : married/widowed A95 : female : single

Attribute 10: (qualitative) Other debtors / guarantors

A101 : none A102 : co-applicant A103 : guarantor

Attribute 11: (numerical) Present residence since

Attribute 12: (qualitative) Property

A121 : real estate A122 : if not A121 : building society savings agreement/ life insurance

A123 : if not A121/A122 : car or other, not in attribute 6 A124 : unknown / no property

Attribute 13: (numerical) Age in years

Attribute 14: (qualitative) Other installment plans

A141 : bank A142 : stores A143 : none

Attribute 15: (qualitative) Housing

A151 : rent A152 : own A153 : for free

Attribute 16: (numerical) Number of existing credits at this bank

Attribute 17: (qualitative) Job

A171 : unemployed/ unskilled - non-resident A172 : unskilled - resident

A173 : skilled employee / official A174 : management/ self-employed/ highly qualified employee

Attribute 18: (numerical) Number of people being liable to provide maintenance for

Attribute 19: (qualitative) Telephone

A191 : none A192 : yes, registered under the customers name

Attribute 20: (qualitative) foreign worker

A201 : yes A202 : no

Cost Matrix This dataset requires use of a cost matrix (see below)

	1	2
1	0	1
2	5	0

(1 = Good, 2 = Bad) the rows represent the actual classification and the columns the predicted classification. It is worse to class a customer as good when they are bad (5), than it is to class a customer as bad when they are good (1).

THE GERMAN CREDIT DATA:

Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. In spite of the fact that the data is German, you should probably make use of it for this assignment.

A few notes on the German dataset:

- DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian.
- Owns_telephone: German phone rates are much higher that in Canada. So fewer people own telephone.

- Foreign_worker: There are million of these in Germany (many from Turkey).It is very hard to get
- German citizenship if you were not born of Germany parents.
- There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into own of two categories. Good or bad.

Task-1

List all the categorical (or nominal) attributes and the real-valued (or numeric) attributes separately.

Solution: From *credit-g.arff* which is a dataset available for the tasks, we can list the following

Number of Instances: 1000

Number of Attributes: 20 (7 numerical, 13 categorical)

Numerical Attributes:

1. Duration 2. Credit_amount 3. Installment_commitment 4. Residece_since

5. Age 6. Existing_credits 7. Num_dependents

Categorical/ nominal attributes:

1. Checking_status 2.Credit_history 3.Purpose 4.Savings_status 5.Employment

6. Personal_status 7.Other_parties 8.Property_magnitude 9.Other_payment_plans

10. Housing 11.Job 12.own_telephone 13.foreign_worker

Task-2

What attributes do you think might be crucial in make the credit assessment? Come up with some simple rules in plain English using your selected attributes.

Solution:

We have attribute Relevancy Analysis by which we can come to know the set of attributes having highest relevancy to the entire data set. We have various popular measures to do this process like

InfoGain, Gini Index,Gain Ratio. Using weka tool we can easily identify the attribute relevancy analysis.

Steps: weka explorer->preprocess->open file(credit-g.arff)->select all->click on select attributes->select attribute evaluator(Info Gain)->start

In the right window we can see the attribute relevancy score generated by this InfoGain measure. The following is the results shown by the InfoGain measure.

Evaluator: weka.attributeSelection.InfoGain Attribute Eval

Search: weka. attribute Selection. Ranker -T -1.7976931348623157E308 -N -1

Relation: german_credit

Instances: 1000

Attributes: 21

Evaluation mode: evaluate on all training data

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

Search Method: Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 21 class):

Information Gain Ranking Filter

Ranked attributes:

0.094739 1 checking_status

0.043618 3 credit_history

0.0329 2 duration

0.028115 6 savings_status

0.024894 4 purpose

0.018709 5 credit_amount

0.016985 12 property_magnitude

0.013102 7 employment

0.012753 15 housing

0.011278 13 age

0.008875 14 other_payment_plans

0.006811 9 personal_status

0.005823 20 foreign_worker

0.004797 10 other_parties

0.001337 17 job

0.000964 19 own_telephone

0 18 num_dependents

0 8 installment_commitment

0 16 existing_credits

0 11 residence_since

Observations:

From the above analysis we can say that the attributes like checking_status, credit_history, duration, savings_status, purpose and credit_amount are ranked high when compared to the other attributes in the dataset.

Task-3

One type of model that you can create is a Decision Tree- Train a decision tree using the complete dataset as the training data. Report the model obtained after training.

Solution:

Steps: weka explorer->preprocess->open file(credit-g.arff)->select all->select classify menu->select classifier J48->se l ect test opti on as “U se trai ni ng set” ->click on start

Now you can see the results in the right window about the classification done through J48

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: german_credit

Instances: 1000

Attributes: 21

Test mode: 10-fold cross-validation

Number of Leaves : 103

Size of the tree : 140

Time taken to build model: 0.17 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	705	70.5 %
Incorrectly Classified Instances	295	29.5 %

Kappa statistic	0.2467
Mean absolute error	0.3467
Root mean squared error	0.4796
Relative absolute error	82.5233 %
Root relative squared error	104.6565 %
Coverage of cases (0.95 level)	92.8 %
Mean rel. region size (0.95 level)	91.7 %

Total Number of Instances 1000

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.84	0.61	0.763	0.84	0.799	0.639	good
	0.39	0.16	0.511	0.39	0.442	0.639	bad
Weighted Avg.	0.705	0.475	0.687	0.705	0.692	0.639	

Note: The decision tree cam also be represented but as we have selected 20 attributes the size of tree is very big in its structure.

Task-4

Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly?(this is also called testing on the training set?) Why do you think you cannot get 100% training accuracy?

Solution:

=== Summary ===

Correctly Classified Instances	705	70.5 %
Incorrectly Classified Instances	295	29.5 %
Kappa statistic	0.2467	
Mean absolute error	0.3467	
Root mean squared error	0.4796	
Relative absolute error	82.5233 %	
Root relative squared error	104.6565 %	
Coverage of cases (0.95 level)	92.8 %	
Mean rel. region size (0.95 level)	91.7 %	

Total Number of Instances 1000

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.84	0.61	0.763	0.84	0.799	0.639	good
	0.39	0.16	0.511	0.39	0.442	0.639	bad
Weighted Avg.	0.705	0.475	0.687	0.705	0.692	0.639	

=== Confusion Matrix ===

a	b	<-- classified as
588	112	a = good
183	117	b = bad

Results/Observations:

The accuracy of classifier depends on several factors like number of selected attributes, filters applied over the dataset, number of cross validations used and finally the size of the data set. In our experiment we got only 70.5% of correct classification by this decision tree method. When we test on different datasets we can com e to certai n concl usi on that 100% accuracy can't b e achieved through any kind of classification method.

Task-5

Is testing on the training set as you did above a good idea? Why or why not?

Solution: Every classifier has two stages

1. Learning
2. Classification.

The method followed in task-3 and task-4 is called as learning. To test the accuracy of classifier we need to supply new test data set where class label is not present to the classifier. Based on the results of classes the accuracy can be measured.

To use the second stage of classifier first we need to create a test data set in the *.arff*

format and can be supplied to the classifier.

Task-6

one approach for solving the problem encountered in the previous question is using cross-validation? Describe what is cross validation briefly? Train a decision tree again using crossvalidation and report your results. Does your accuracy increase/decrease? Why?

Solution: Cross validatation is a common technique for assessing classifier accuracy, based on randomly sampled partitions of the given data. In k-fold corss- validation the initial data are randomly porti oned i nto k m utual l y ex cl usi v e subsets or “fol ds”, s1,s 2/ . . sk, each of approxi m atel y equal si ze. Training and testing is performed k times. In iteration I, the subset Si is reserved as the test set, and the remaining subset are collectively used to train the classifier. That is the classifier of the first iteration is trai ned on subsets s2,/ . . sk and tested on S2 and so on . The accuracy esti m ate i s the ov eral l num ber of correct classifications from the k iterations, divided by the total number of samples in the initial data.

In weka tool using cross validation testing method, first once the dataset is been given to the tool using explorer and select al l attri butes of the dataset usi ng “! LL” but ton and select Classify menu option, select classifier as “J 48”, U se the test opti on as cross v al i dati on where fol ds wi ll be 10 by defaul t selection by weka and click on start and we can see the results of this method.

Here accuracy of this method using the same data set when compared to decision tree does not vary. We can try out for various combinations of this option of cross validation with different folding values like 2, 5, 8, 10(default), 15 20 50 and see the accuracy. Finally with this method we can come to a conclusion that in cross validation the folds should not be varies neither high nor very low.

Task-7

Check to see if the data shows a bias a gainst “foreign workers” (attribute 20), OR “personal status” (attribute 9). One way to do this (perhaps rather simple method) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in we ka’s GUI Explore r. Did re moving these attribute s have any significant effect? Discuss.

Solution: To check whether the two attributes like foreign_workers and personal_status

are biased by the dataset, first those two attributes are removed from the selection of the dataset by the weka explorer using the option " remove". after the classification through the decision tree we can come to certain conclusion that these attributes does not place any major role on the data set for it s classification.

In other words we can also say that these attribute whose relevant score is very low which is generated through weka tool using the measure called InforGain.

Task-8

Another question might be, do you really need to input so many attributes to get good results? May be only few world do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17(and 21, the class attribute (naturally)).Try out some combinations.

Solution: For the data classification using the tool like weka if we supply the entire dataset attributes to the tool the result generated by the tool in the form of decision tree is looking like a diagram which we can't understand in a single snapshot and can' t understand the complete classification generated by this method.

To avoid the above problems we need to supply the attributes of the dataset which are having high relevancy score. Select top three or four attributes which we got through InfoGain measure in the weka tool and see the decision tree generated in this method which is understandable and easy to draw and analyze.

Results from Weka:

The following is the decision tree by supplying four attributes to the classifier algorithm.