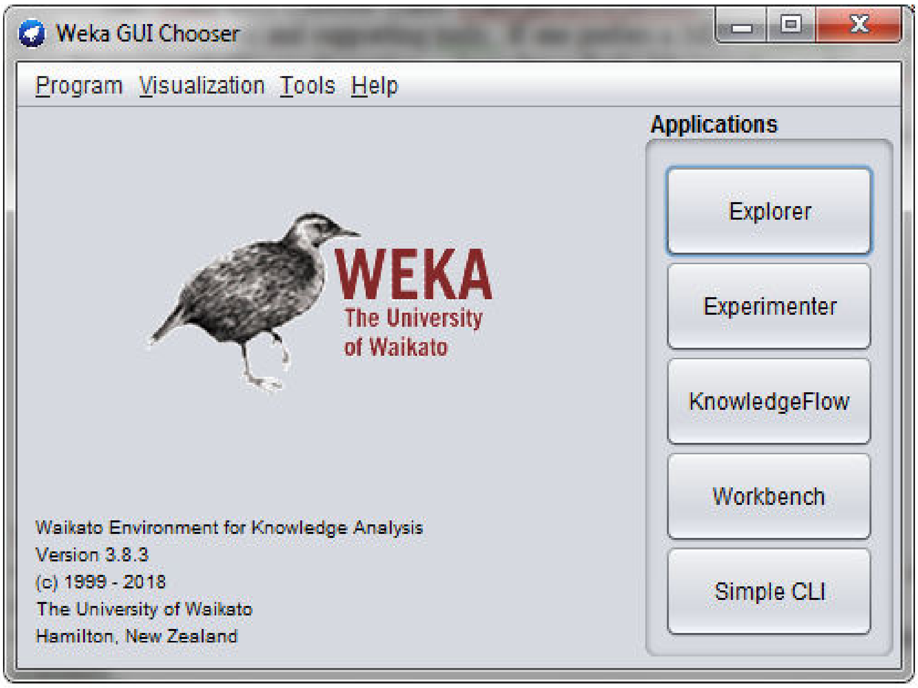
Add-on Experiment: A

Aim :- Study of WEKA Tool

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 interface") appearance, then this is provided by an alternative launcher called " Ma in" (class weka.gui.Main).

The GUI Chooser consists of Five Buttons — One for each of the Five major Weka applications — and Four Menus.



Waikato Environment for Knowledge Analysis
Version 3.8.3
(c) 1999 - 2018
The University of Waikato
Hamilton, New Zealand

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.
- Knowledge Flow: 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.
- Workbench: This consists of threshold values for all applications of Weka.

- Simple CLI 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 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 greyed 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.

Status Box

The status box appears at the very bottom of the window. It displays messages that keep you informed about what’s going on. For example, if the Explorer is busy loading a file, the status box will say that.

TIP—right-clicking the mouse anywhere inside the status box brings up a little menu. The menu gives two options:

1. Memory information: Display in the log box the amount of memory available to WEKA.
2. Run garbage collector: Force the Java garbage collector to search for memory that is no longer needed and free it up, allowing more memory for new tasks. Note that the garbage collector is constantly running as a back ground task any way.

Log Button

Clicking on this button brings up a separate window containing a scrollable text file ld. Each line of text is stamped with the time it was entered into the log. As you perform actions in WEKA, the log keeps a record of what has happened.

WEKA Status Icon

To the right of the status box is the WEKA status icon. When no processes are running, the bird sits down and takes a nap. The number beside the × symbol gives the number of concurrent processes running. When the system is idle it is zero, but it increases as the number of processes increases. When any process is started, the bird gets up and starts moving around. If it’s standing but stops moving for a long time, it’s sick: something has gone wrong! In that case you should restart the WEKA Explorer.

Graphical output

Most graphical displays in WEKA, e.g., the Graph Visualizer or the Tree Visualizer, support saving the output to a file. A dialog for saving the output can be brought up with Alt+Shift+left-click. Supported formats are currently Windows Bitmap, JPEG, PNG and EPS (encapsulated Postscript). The dialog also allows you to specify the dimensions of the generated image.

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 data base. (Note that to make this work you might have to edit the file in weka / experiment/ Database Utils.props .)
4. Generate . . . . ..Enables you to generate artificial data from a variety of Data Generators .

The Current Relation

1. Relation. The name of the relation, as given in the file it was loaded from. Filters (described below) modify the name of a relation.
2. Instances. The number of instances (data points / records) in the data.
3. Attributes. The number of attributes (features) in the data.

Working with Attributes: - Below the current relation box is a box titled Attributes. There are four buttons and beneath them is a list of the attributes in the current relation. The list has three columns:

1. **NO**. A number that identifies the attribute in the order they are specified in the data file.
2. **Selection tick boxes**. These allow you select the attributes are present in the relation.
3. **Name.** The name of the attribute as it was declared in the data file.

When you click on different rows in the list of attributes, the fields change in the box to the right titled selected attribute. This box displays the characteristics of the currently highlighted attribute in the list:

1. **Name.** The name of the attribute the same as that given in the attribute list.
2. **Type.** The type of attribute most commonly Nominal or Numeric.
3. **Missing.** The number of instances in the data for which this attribute is missing (unspecified).
4. **Distinct.** The number of different values that the data contains for this attribute.

5. **Unique**. The number of instances in the data having a value for this attribute that no other instances have.

Returning to the attribute list to begin with all the tick boxes are un ticked. They can be toggled on/off by clicking on them individually. The four buttons above can also be used to change the selection:

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.

Add-on Experiment: B

Aim: - Create a data file in ARFF format.

Problem Statement: - Create a data file in ARFF format manually using any editor such as notepad or turbo editor.

Theory: - An ARFF (Attribute-relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes .ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software.

ARFF files have two distinct sections, **Header** information and **Data** information. The **Header** of the ARFF file contains the name of the relation, a list of the attributes (the columns in the data), and their types. The ARFF Header section of the file contains the relation declaration and attributes declarations.

The @relation Declaration:-The relation name is defined as the first line in the ARFF file, The Format is:@relation<relation-name> is a string, the string must be quoted if the name includes spaces.

Examples:

@RELATION iris
@RELATION bank

The @attribute Declaration:-Attribute Declarations take the form of an ordered sequence of @attribute statements. Each attribute in the data set has its own @attribute statement which uniquely defines the name of that attribute is the third one declared then Weka expects that all that attributes values will be found in the third comma delimited column. The format for the @attribute statement is:@attribute <attribute-name><data type>,where the <attribute-name> must start with an alphabetic character .If spaces are to be included in the name then the entire name must be quoted. The <data type>can be any of the four types currently(version 3.2.1)supported by Weka. Numeric ,<nominal-specification>,string, date[<date-format>].

Numeric attributes can be real or integer numbers. Nominal attributes :-Nominal values are defined by providing an <nominal-specification>listing the possible values:{<nominal-name1>,<nominal-name2>,<nominal-name3>,…..} For example, @ATTRIBUTE gender {male, female}.String attributes allow us to create attributes containing arbitrary textual values. Date attribute declarations take the form :@attribute <name> date[<date-format>] where <name> is the name for the attribute ,<date-format> is an optional string specifying how date values should be parsed and printed (this is the same format used by SimpleDateFormat).

The default format string accepts the ISO-8601 combined date and time format:”yyyy-MM-dd`T`HH:mm:ss”.

Examples :

@ATTRIBUTE petalwidth NUMERIC
@ATTRIBUTE class {First,Second,Third}
@ATTRIBUTE name STRING
@ATTRIBUTE dob DATE
@ATTRIBUTE doj DATE “yyyy-MM-dd HH:mm:ss”.

ARFF Data Section:-

The ARFF Data section of the file contains the data declaration line and the actual instance lines. The @data declaration is a single line denoting the start of the data segment in the file. The format is:@data .Each instance is represented on a single line; with carriage returns denoting the end of the instance .Attribute values for each instance are delimited by comas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute).

Examples

@DATA 5.1,3.5,1.4,0.2,Iris-setosa

Sample input :-
<u>Stu.arff</u>

```
@relation stu
@attribute sno
@attribute sname
@attribute  gender{male, female}
@data
1,sri,male
2,ashok,male
3,vinay,male
4,chinni,female
```

Output :-ARFF file successfully created. We can see the result by opening the file with WEKA tool.

Add-on Experiment: C

Aim: **-** Transforming the Excel data set into ARFF file data and also Verifying with WEKA tool.

Procedure:-

1. Convert the Excel data set into CSV (comma separated value) format.
2. One easy way to do this is to load it into Excel and use “Save AS” to save the file in CSV format.
3. Edit the CSV file , and add the ARFF header information to the file.
4. This involves creating the @relation line , one @attribute line per attribute and @data to signify the start of data.
5. Finally save this file with (.arff) extension.
6. It is also considered good practice to add comments at the top of the file.
7. Describing where you obtained this data set, what its summary characteristics are , etc.
8. A comment in the ARFF format is started with the percent character % and continues until the end of the line.
9. Open this file with WEKA tool.

Sample Input:-

1. Create an excel file named vvitit.csv

AJWT	DM	HCI
28	32	33
32	30	30
33	35	28
34	36	29
35	37	30

36	24	31
24	25	32
25	26	33
26	27	34
27	28	29
28	29	30
29	30	31
30	31	32
26	23	33
27	23	34
28	30	35
29	31	21
30	34	31
31	35	30

2. Select the file and open with notepad option

```
@relation convert
@attribute AJWT numeric
@attribute DM numeric
@attribute HCI numeric
@attribute STATUS {p,f}
@data
28,32,33,p
32,30,30,p
33,35,28,p
34,36,29,p
35,37,30,p
36,24,31,f
24,25,32,f
25,26,33,f
26,27,34,p
27,28,29,p
28,29,30,f
29,30,31,p
30,31,32,p
26,23,33,f
27,23,34,p
28,30,35,p
29,31,21,f
30,34,31,p
31,35,30,p
```

Output :

Successfully Excel file converted into ARFF file format and we can see the result by opening the file with WEKA tool.

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 strategies 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. The 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 partition 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}
```

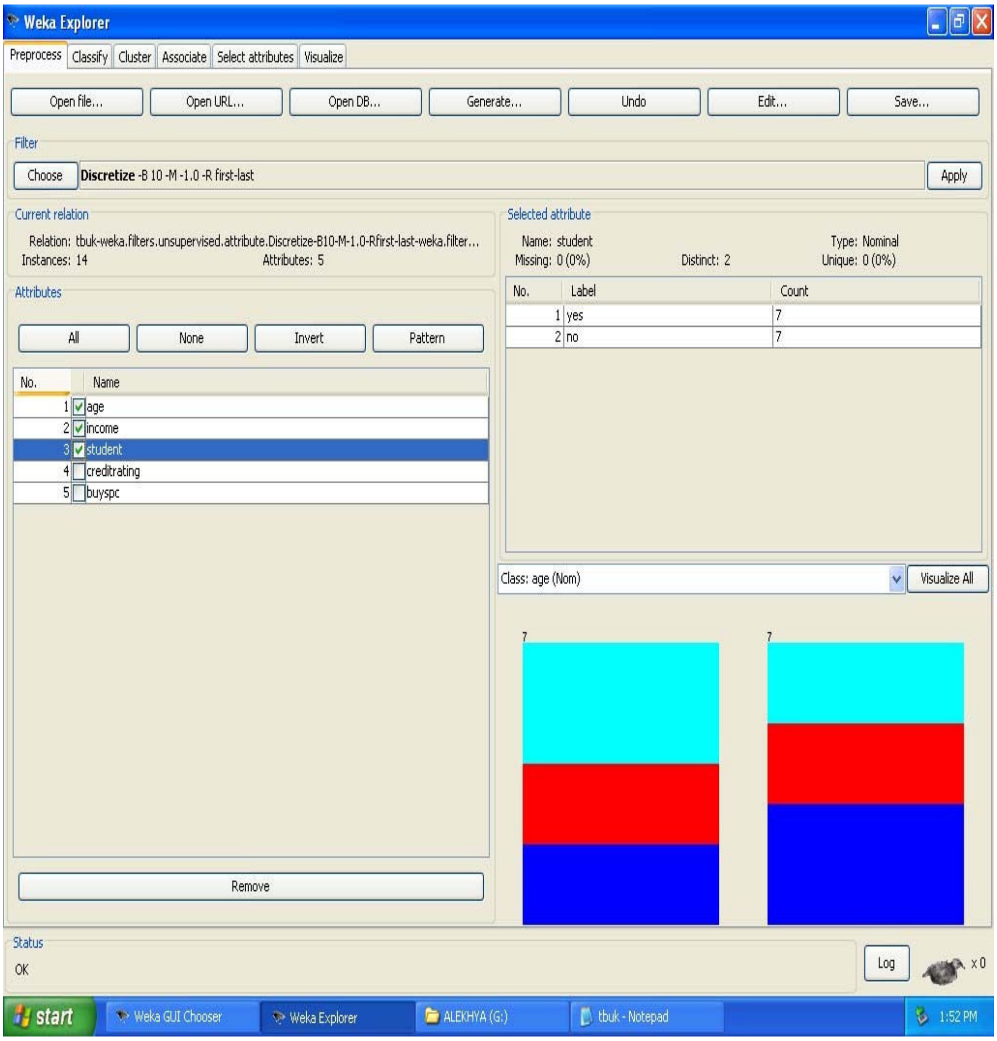
```
@data

%

<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.



Filter: **Discretize** -B 10 -M -1.0 -R first-last

Current relation

Relation: tbuk-weka.filters.unsupervised.attribute.Discretize-B10-M-1.0-Rfirst-last-weka.filter...
Instances: 14 Attributes: 5

Attributes

No.		Name
1	☑	age
2	☑	income
3	☑	student
4	□	creditrating
5	□	buyspc

Selected attribute

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

No.	Label	Count
1	yes	7
2	no	7

Class: age (Nom)

Status
OK