Machine Learning with WEKA

WEKA Explorer Tutorial

for WEKA Version 3.4.3

Svetlana S. Aksenova
aksenovs@ecs.csus.edu

School of Engineering and Computer Science
Department of Computer Science
California State University, Sacramento
California, 95819

2004

TABLE OF CONTENTS

1. Introduction

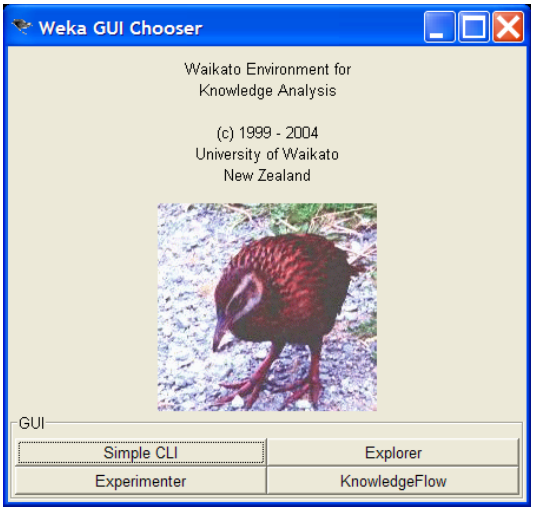
WEKA is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. WEKA is a state-of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning algorithms for data mining tasks. The algorithms are applied directly to a dataset. WEKA implements algorithms for data preprocessing, classification, regression, clustering, association rules; it also includes a visualization tools. The new machine learning schemes can also be developed with this package. WEKA is open source software issued under the GNU General Public License [3].

The goal of this Tutorial is to help you to learn WEKA Explorer. The tutorial will guide you step by step through the analysis of a simple problem using WEKA Explorer preprocessing, classification, clustering, association, attribute selection, and visualization tools. At the end of each problem there is a representation of the results with explanations side by side. Each part is concluded with the exercise for individual practice. By the time you reach the end of this tutorial, you will be able to analyze your data with WEKA Explorer using various learning schemes and interpret received results.

Before starting this tutorial, you should be familiar with data mining algorithms such as C4.5 (C5), ID3, K-means, and Apriori. All working files are provided. For better performance, the archive of all files used in this tutorial can be downloaded or copied from CD to your hard drive as well as a printable version of the lessons. A trial version of Weka package can be downloaded from the University of Waikato website at http://www.cs.waikato.ac.nz/~ml/weka/index.html.

2. Launching WEKA Explorer

You can launch Weka from C:\Program Files directory, from your desktop selecting Weka-3-4 Shortcut 2 KB icon, or from the Windows task bar 'Start' → 'Programs' → 'Weka 3-4'. When 'WEKA GUI Chooser' window appears on the screen, you can select one of the four options at the bottom of the window [2]:



Weka GUI Chooser

Waikato Environment for
Knowledge Analysis

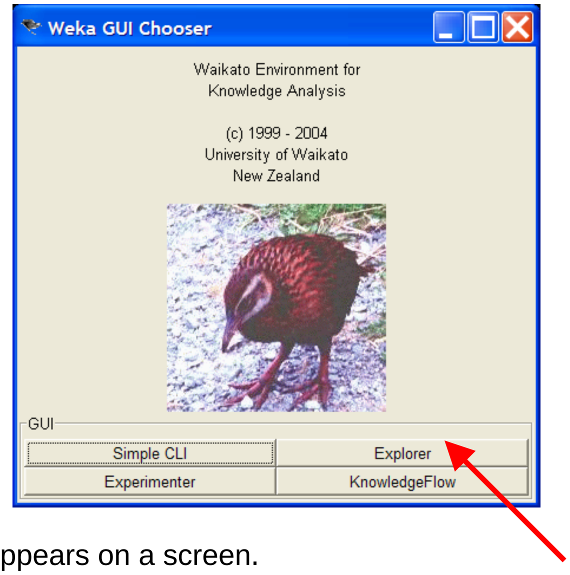
(c) 1999 - 2004
University of Waikato
New Zealand

1. **Simple CLI** provides a simple command-line interface and allows direct execution of Weka commands.

2. **Explorer** is an environment for exploring data.

3. **Experimenter** is an environment for performing experiments and conducting statistical tests between learning schemes.
4. **KnowledgeFlow** is a Java-Beans-based interface for setting up and running machine learning experiments.

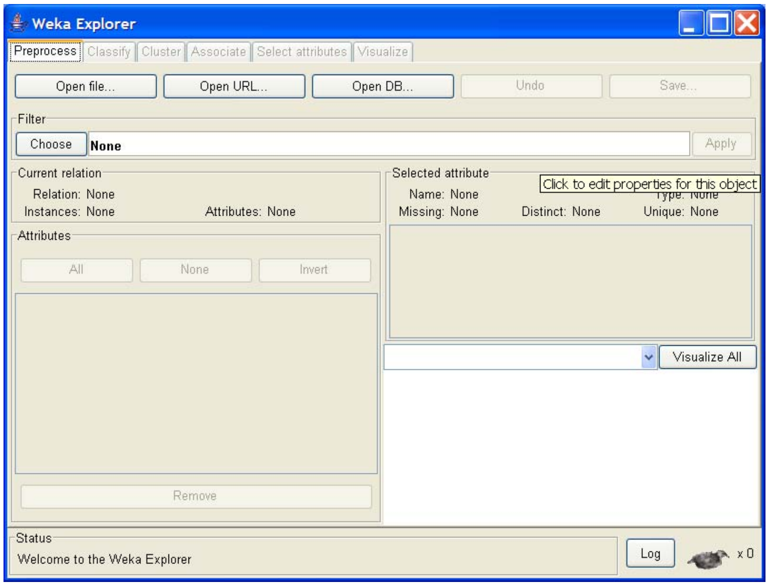
For the exercises in this tutorial you will use ‘Explorer’. Click on ‘Explorer’ button in the ‘WEKA GUI Chooser’ window.



Waikato Environment for
Knowledge Analysis

(c) 1999 - 2004
University of Waikato
New Zealand

‘WEKA Explorer’ window appears on a screen.



Filter: None

Relation: None
Instances: None
Attributes: None

Name: None
Type: None
Missing: None
Distinct: None
Unique: None

Click to edit properties for this object

Welcome to the Weka Explorer

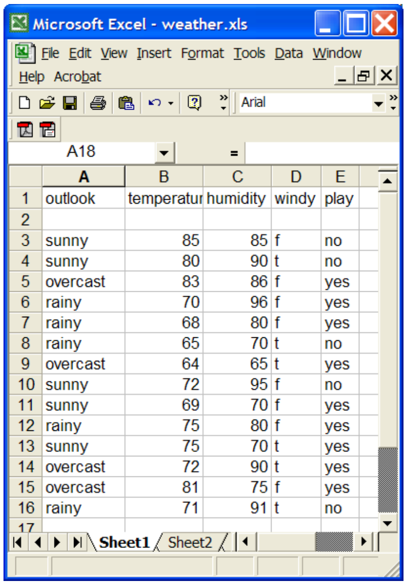
3. Preprocessing Data

At the very top of the window, just below the title bar there is a row of tabs. Only the first tab, ‘Preprocess’, is active at the moment because there is no dataset open. The first three buttons at the top of the preprocess section enable you to load data into WEKA. Data can be imported from a file in various formats: ARFF, CSV, C4.5, binary, it can also be read from a URL or from an SQL database (using JDBC) [4]. The easiest and the most common way of getting the data into WEKA is to store it as Attribute-Relation File Format (ARFF) file.

You've already been given “weather.arff” file for this exercise; therefore, you can skip section 3.1 that will guide you through the file conversion.

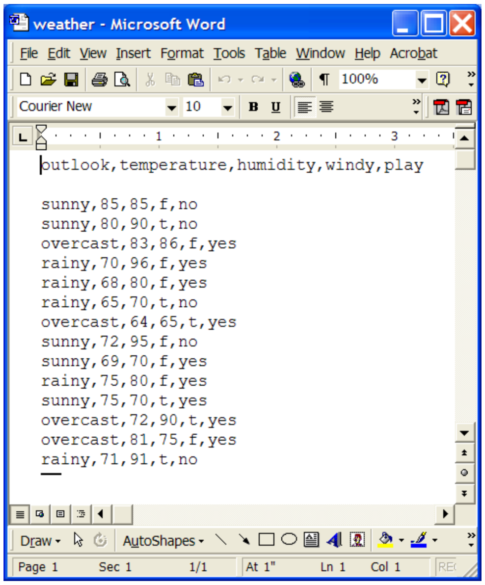
3.1. File Conversion

We assume that all your data stored in a Microsoft Excel spreadsheet “weather.xls”.


Microsoft Excel - weather.xls
File Edit View Insert Format Tools Data Window
Help Acrobat
Arial
A18
=
Sheet1
Sheet2

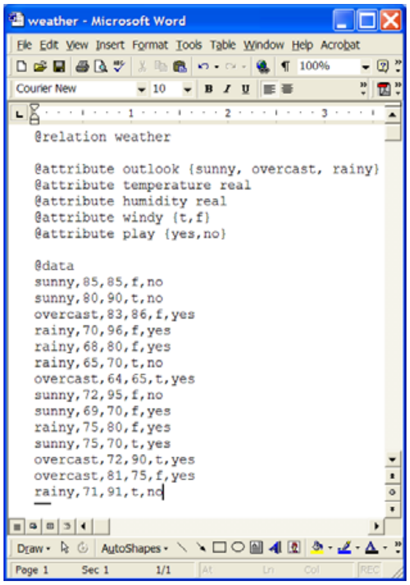
	A	B	C	D	E
1	outlook	temperatur	humidity	windy	play
2					
3	sunny	85	85	f	no
4	sunny	80	90	t	no
5	overcast	83	86	f	yes
6	rainy	70	96	f	yes
7	rainy	68	80	f	yes
8	rainy	65	70	t	no
9	overcast	64	65	t	yes
10	sunny	72	95	f	no
11	sunny	69	70	f	yes
12	rainy	75	80	f	yes
13	sunny	75	70	t	yes
14	overcast	72	90	t	yes
15	overcast	81	75	f	yes
16	rainy	71	91	t	no
17					

WEKA expects the data file to be in Attribute-Relation File Format (ARFF) file. Before you apply the algorithm to your data, you need to convert your data into comma-separated file into ARFF format (into the file with .arff extension) [1]. To save you data in comma-separated format, select the ‘Save As…’ menu item from Excel ‘File’ pull-down menu. In the ensuing dialog box select ‘CSV (Comma Delimited)’ from the file type pop-up menu, enter a name of the file, and click ‘Save’ button. Ignore all messages that appear by clicking ‘OK’. Open this file with Microsoft Word. Your screen will look like the screen below.


weather - Microsoft Word
File Edit View Insert Format Tools Table Window Help Acrobat
100%
Courier New 10
Draw AutoShapes
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC

```
outlook,temperature,humidity,windy,play

sunny,85,85,f,no
sunny,80,90,t,no
overcast,83,86,f,yes
rainy,70,96,f,yes
rainy,68,80,f,yes
rainy,65,70,t,no
overcast,64,65,t,yes
sunny,72,95,f,no
sunny,69,70,f,yes
rainy,75,80,f,yes
sunny,75,70,t,yes
overcast,72,90,t,yes
overcast,81,75,f,yes
rainy,71,91,t,no
```

The rows of the original spreadsheet are converted into lines of text where the elements are separated from each other by commas. In this file you need to change the first line, which holds the attribute names, into the header structure that makes up the beginning of an ARFF file. Add a `@relation` tag with the dataset's name, an `@attribute tag with` the attribute information, and a `@data` tag as shown below.

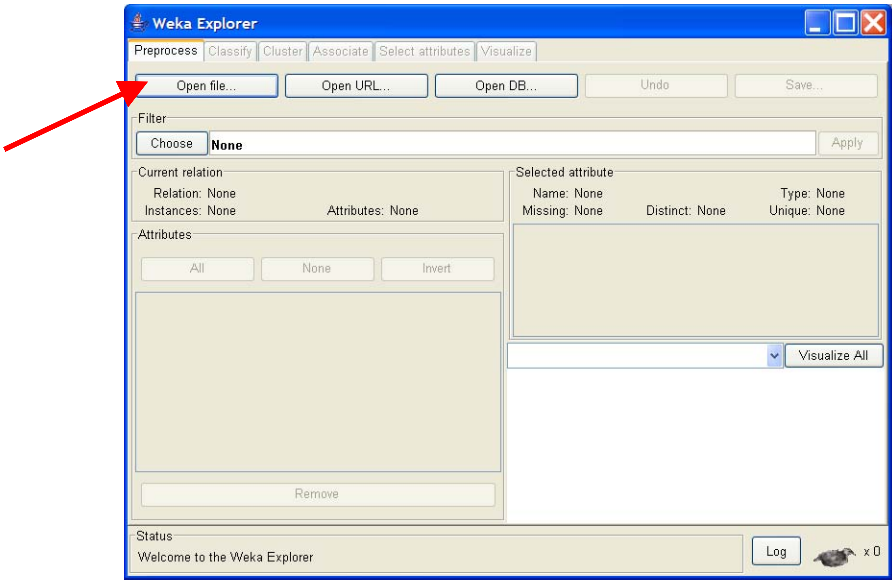

weather - Microsoft Word
File Edit View Insert Format Tools Table Window Help Acrobat
100%
Courier New 10
Draw AutoShapes
Page 1 Sec 1 1/1 At Ln Col REC

```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {t,f}
@attribute play {yes,no}

@data
sunny,85,85,f,no
sunny,80,90,t,no
overcast,83,86,f,yes
rainy,70,96,f,yes
rainy,68,80,f,yes
rainy,65,70,t,no
overcast,64,65,t,yes
sunny,72,95,f,no
sunny,69,70,f,yes
rainy,75,80,f,yes
sunny,75,70,t,yes
overcast,72,90,t,yes
overcast,81,75,f,yes
rainy,71,91,t,no
```

Choose 'Save As…' from the 'File' menu and specify 'Text Only with Line Breaks' as the file type. Enter a file name and click 'Save' button. Rename the file to the file with extension .arff to indicate that it is in ARFF format.

3.2. Opening file from a local file system

Click on 'Open file…' button.



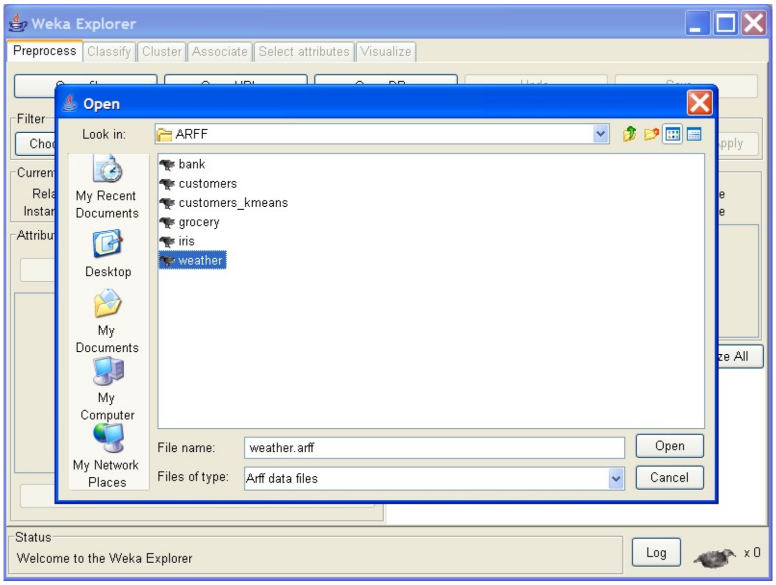
Filter: None

Current relation
Relation: None
Instances: None
Attributes: None

Selected attribute
Name: None
Type: None
Missing: None
Distinct: None
Unique: None

Status
Welcome to the Weka Explorer

It brings up a dialog box allowing you to browse for the data file on the local file system, choose "weather.arff" file.



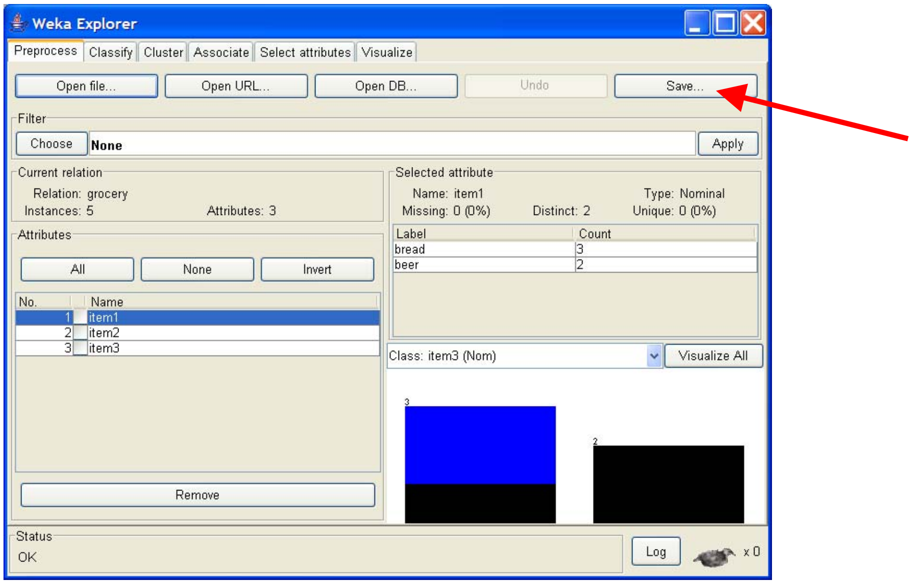
Look in: ARFF

File name: weather.arff

Files of type: Arff data files

Status
Welcome to the Weka Explorer

Some databases have the ability to save data in CSV format. In this case, you can select CSV file from the local filesystem. If you would like to convert this file into ARFF format, you can click on 'Save' button. WEKA automatically creates ARFF file from your CSV file.



Filter: None

Current relation
Relation: grocery
Instances: 5 Attributes: 3

Attributes

No.		Name
1	□	item1
2	□	item2
3	□	item3

Selected attribute
Name: item1 Type: Nominal
Missing: 0 (0%) Distinct: 2 Unique: 0 (0%)

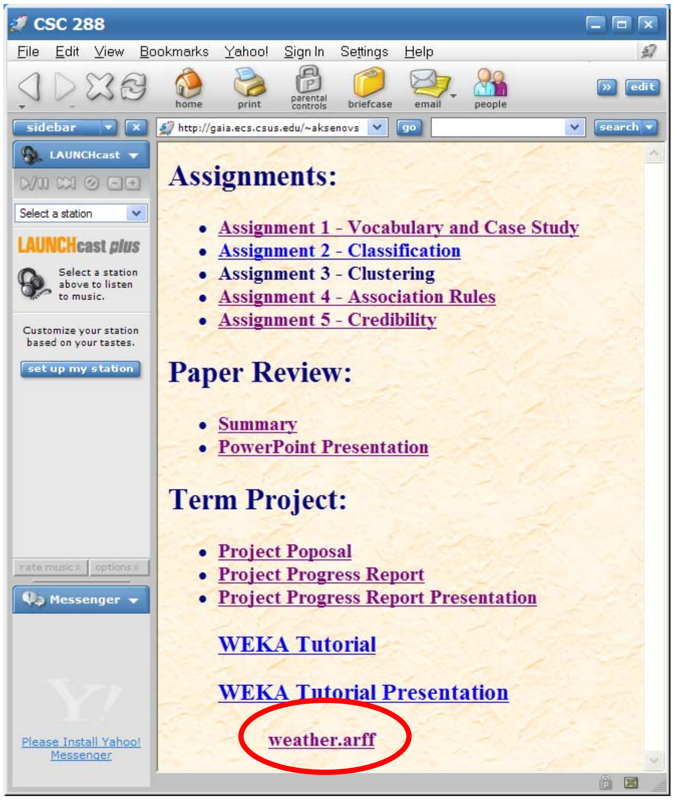
Label	Count
bread	3
beer	2

Class: item3 (Nom)

Status
OK

3.3. Opening file from a web site

A file can be opened from a website. Suppose, that “weather.arff” is on the following website:



http://gaia.ecs.csus.edu/~aksenovs

Assignments:

- Assignment 1 - Vocabulary and Case Study
- Assignment 2 - Classification
- Assignment 3 - Clustering
- Assignment 4 - Association Rules
- Assignment 5 - Credibility

Paper Review:

- Summary
- PowerPoint Presentation

Term Project:

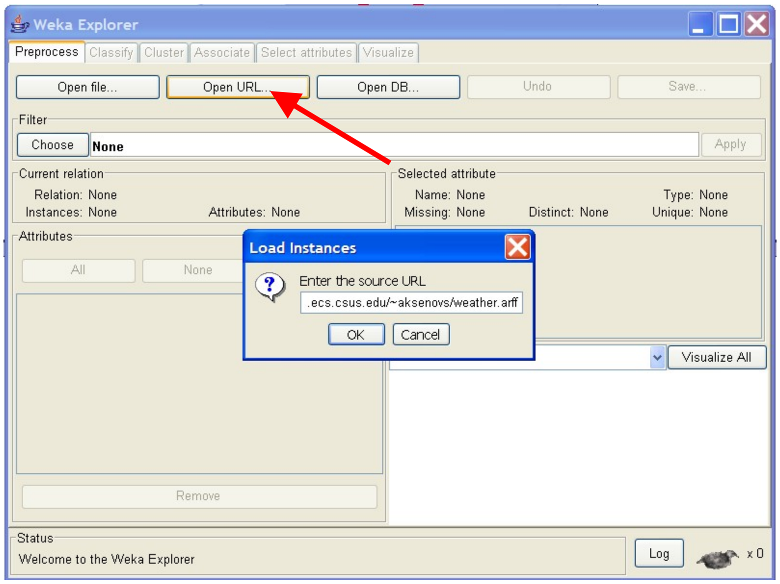
- Project Poposal
- Project Progress Report
- Project Progress Report Presentation

WEKA Tutorial

WEKA Tutorial Presentation

weather.arff

The URL of the web site in our example is http://gaia.ecs.csus.edu/~aksenovs/. It means that the file is stored in this directory, just as in the case with your local file system. To open this file, click on ‘Open URL…’ button, it brings up a dialog box requesting to enter source URL.



Load Instances

Enter the source URL

.ecs.csus.edu/~aksenovs/weather.arff

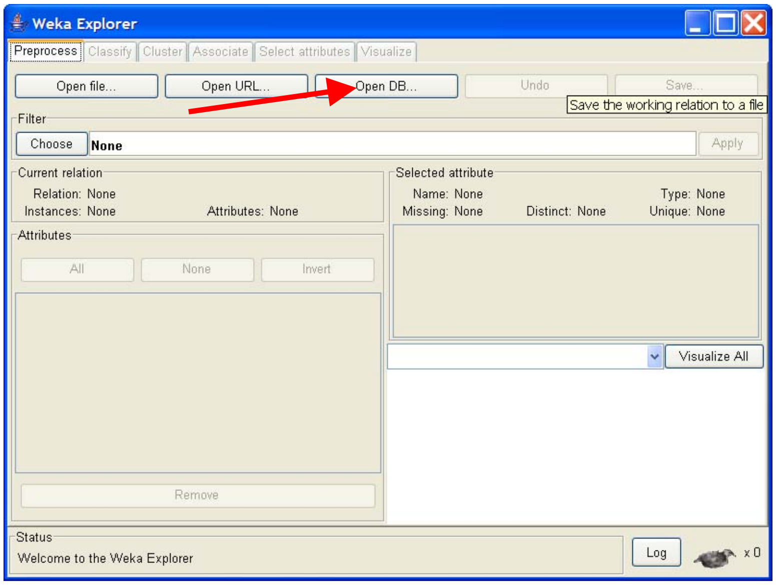
OK Cancel

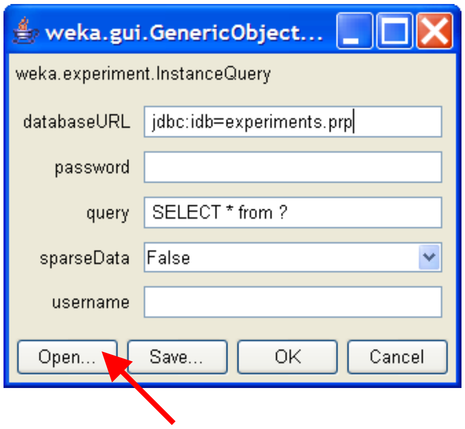
Welcome to the Weka Explorer

Enter the URL of the web site followed by the file name, in this example the URL is http://gaia.ecs.csus.edu/~aksenovs/weather.arff, where weather.arff is the name of the file you are trying to load from the website.

3.4. Reading data from a database

Data can also be read from an SQL database using JDBC. Click on 'Open DB…' button, 'GenericObjectEditor' appears on the screen.





weka.experiment.InstanceQuery

databaseURL jdbc:idb=experiments.prp

password

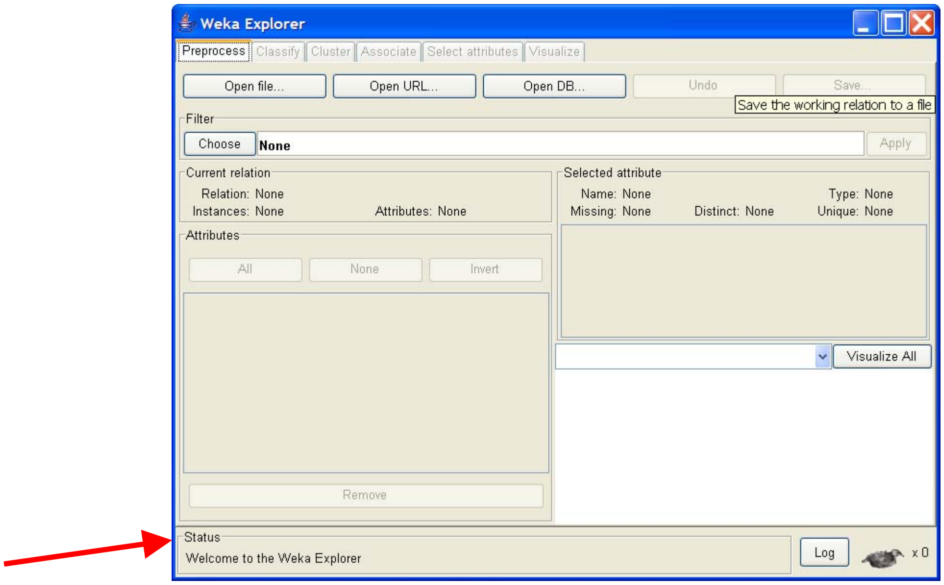
query SELECT * from ?

sparseData False

username

To read data from a database, click on 'Open' button and select the database from a filesystem.

3.5. Preprocessing window



None

Relation: None
Instances: None
Attributes: None

Name: None
Type: None
Missing: None
Distinct: None
Unique: None

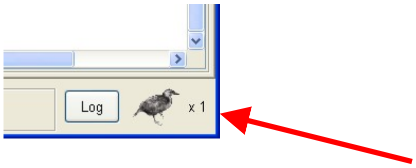
Welcome to the Weka Explorer

At the bottom of the window there is 'Status' box. The 'Status' box displays messages that keep you informed about what is going on. For example, when you first opened the 'Explorer', the message says, "Welcome to the Weka Explorer". When you loading "weather.arff" file, the 'Status' box displays the message "Reading from file…". Once the file is loaded, the message in the 'Status' box changes to say "OK". Right-click anywhere in 'Status box', it brings up a menu with two options:

1. **Available Memory** that displays in the log and in 'Status' box the amount of memory available to WEKA in bytes.
2. **Run garbage collector** that forces Java garbage collector to search for memory that is no longer used, free this memory up and to allow this memory for new tasks.

To the right of 'Status box' there is a 'Log' button that opens up the log. The log records every action in WEKA and keeps a record of what has happened. Each line of text in the log contains time of entry. For example, if the file you tried to open is not loaded, the log will have record of the problem that occurred during opening.

To the right of the 'Log' button there is an image of a bird. The bird is WEKA status icon. The number next to 'X' symbol indicates a number of concurrently running processes. When you loading a file, the bird sits down that means that there are no processes running. The number of processes besides symbol 'X' is zero that means that the system is idle. Later, in classification problem, when generating result look at the bird, it gets up and start moving that indicates that a process started. The number next to 'X' becomes 1 that means that there is one process running, in this case calculation.

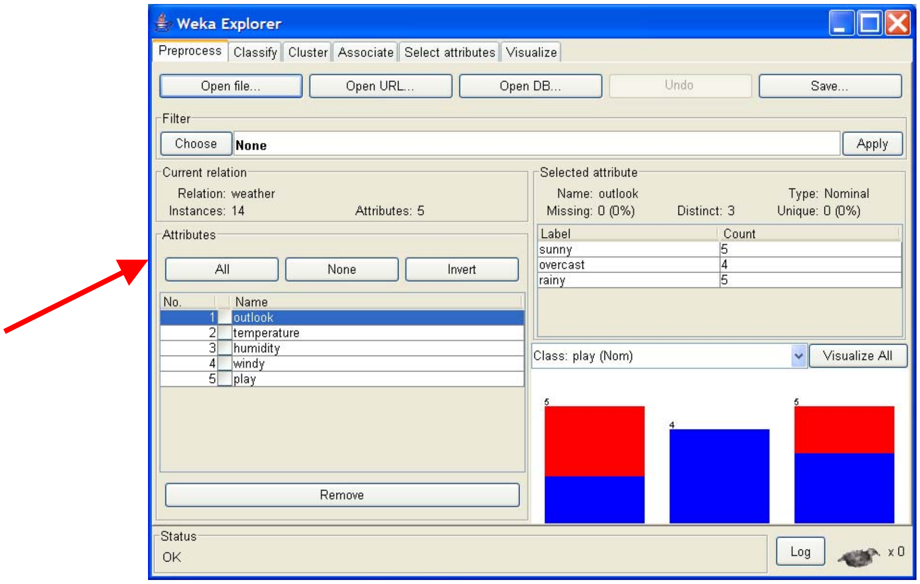


If the bird is standing and not moving for a long time, it means that something has gone wrong. In this case you should restart WEKA Explorer.

Loading data

Lets load the data and look what is happening in the 'Preprocess' window.

The most common and easiest way of loading data into WEKA is from ARFF file, using 'Open file…' button (section 3.2). Click on 'Open file…' button and choose "weather.arff" file from your local filesystem. Note, the data can be loaded from CSV file as well because some databases have the ability to convert data only into CSV format.



Relation: weather
Instances: 14
Attributes: 5

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Name: outlook
Type: Nominal
Missing: 0 (0%)
Distinct: 3
Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

Class: play (Nom)

OK

Once the data is loaded, WEKA recognizes attributes that are shown in the 'Attribute' window. Left panel of 'Preprocess' window shows the list of recognized attributes:

No. is a number that identifies the order of the attribute as they are in data file,
Selection tick boxes allow you to select the attributes for working relation,
Name is a name of an attribute as it was declared in the data file.

The 'Current relation' box above 'Attribute' box displays the base relation (table) name and the current working relation (which are initially the same) - "weather", the number of instances - 14 and the number of attributes - 5.

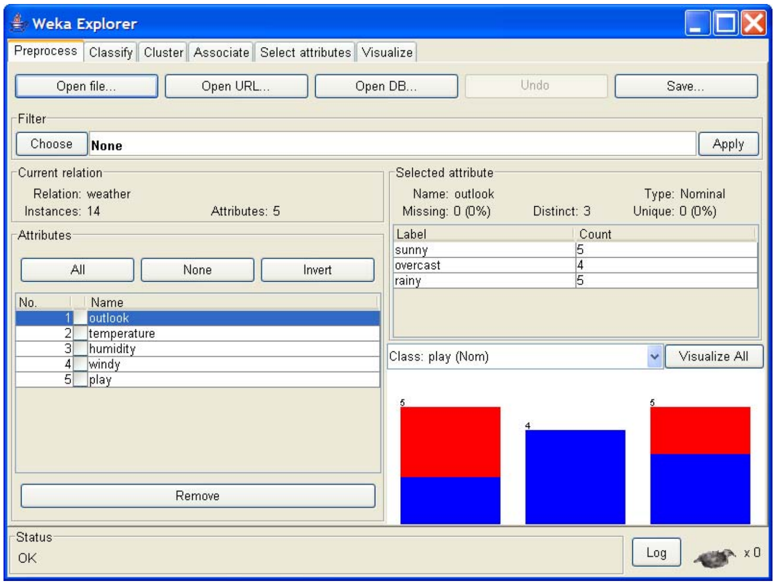
During the scan of the data, WEKA computes some basic statistics on each attribute. The following statistics are shown in 'Selected attribute' box on the right panel of 'Preprocess' window:

Name is the name of an attribute,
Type is most commonly Nominal or Numeric, and
Missing is the number (percentage) of instances in the data for which this attribute is unspecified,
Distinct is the number of different values that the data contains for this attribute, and
Unique is the number (percentage) of instances in the data having a value for this attribute that no other instances have.

An attribute can be deleted from the ‘Attributes’ window. Highlight an attribute you would like to delete and hit Delete button on your keyboard.

By clicking on an attribute, you can see the basic statistics on that attribute. The frequency for each attribute value is shown for categorical attributes. Min, max, mean, standard deviation (StdDev) is shown for continuous attributes.

Click on attribute Outlook in the ‘Attribute’ window.



Filter: None

Current relation
Relation: weather
Instances: 14 Attributes: 5

Selected attribute
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

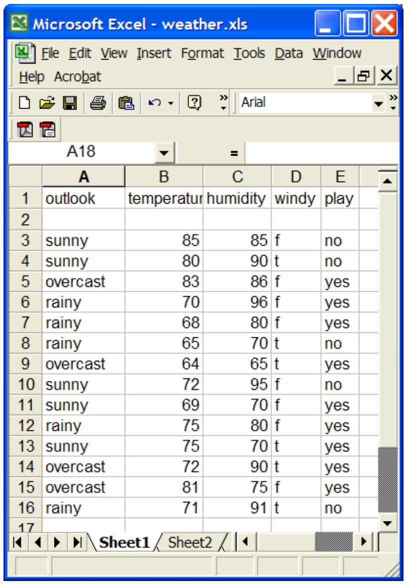
Label	Count
sunny	5
overcast	4
rainy	5

Attributes

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: play (Nom)

Status
OK


Microsoft Excel - weather.xls
File Edit View Insert Format Tools Data Window
Help Acrobat
Arial
Sheet1
Sheet2

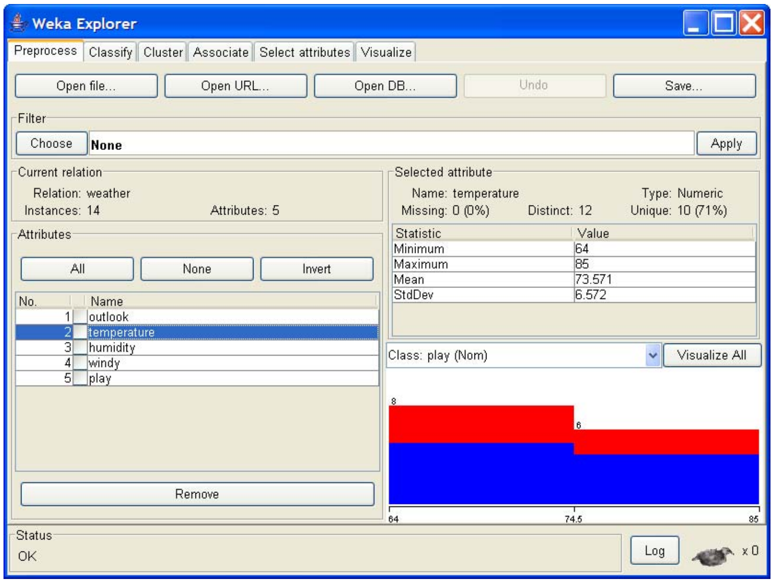
A18 =

	A	B	C	D	E
1	outlook	temperatur	humidity	windy	play
2					
3	sunny	85	85	f	no
4	sunny	80	90	t	no
5	overcast	83	86	f	yes
6	rainy	70	96	f	yes
7	rainy	68	80	f	yes
8	rainy	65	70	t	no
9	overcast	64	65	t	yes
10	sunny	72	95	f	no
11	sunny	69	70	f	yes
12	rainy	75	80	f	yes
13	sunny	75	70	t	yes
14	overcast	72	90	t	yes
15	overcast	81	75	f	yes
16	rainy	71	91	t	no
17					

Outlook is nominal. Therefore, you can see the following frequency statistics for this attribute in the ‘Selected attributes’ window:
Missing = 0 means that the attribute is specified for all instances (no missing values),
Distinct = 3 means that Outlook has three different values: sunny, overcast, rainy, and
Unique = 0 means that other instances do not have the same value as Outlook has.

Just below these values there is a table displaying count of instances of the attribute Outlook. As you can see, there are three values: sunny with 5 instances, overcast with 4 instances, and rainy with 5 instances. These numbers match the numbers of instances in the base relation and table “weather.xls”.

Lets take a look at the attribute Temperature.



Filter: None

Current relation
Relation: weather
Instances: 14 Attributes: 5

Selected attribute
Name: temperature Type: Numeric
Missing: 0 (0%) Distinct: 12 Unique: 10 (71%)

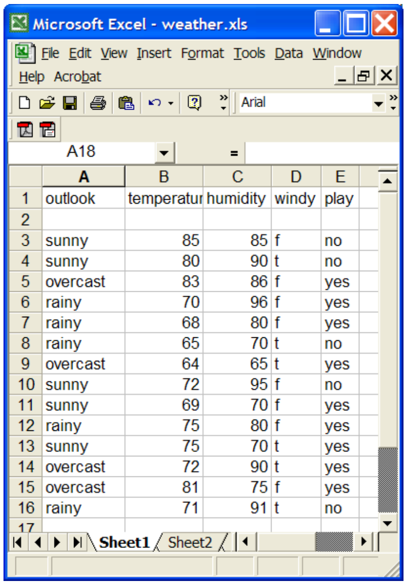
Statistic	Value
Minimum	64
Maximum	85
Mean	73.571
StdDev	6.572

Attributes

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: play (Nom)

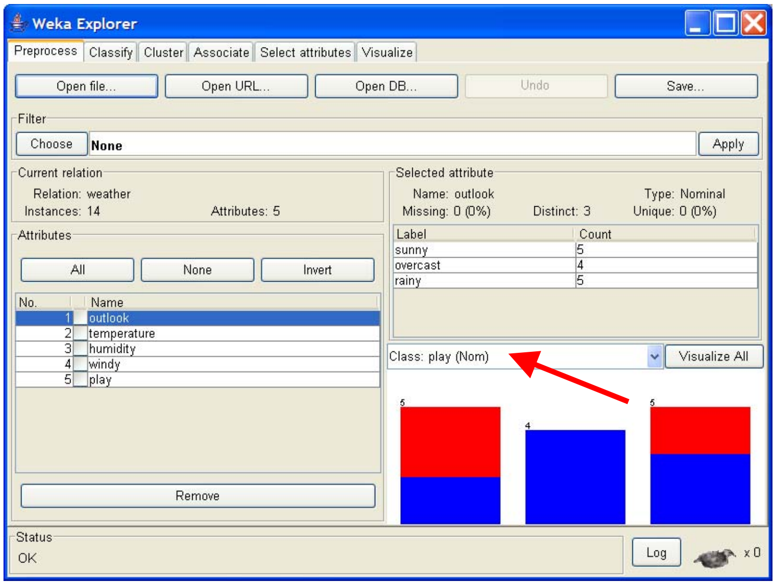
Status
OK



	A	B	C	D	E
1	outlook	temperatur	humidity	windy	play
2					
3	sunny	85	85	f	no
4	sunny	80	90	t	no
5	overcast	83	86	f	yes
6	rainy	70	96	f	yes
7	rainy	68	80	f	yes
8	rainy	65	70	t	no
9	overcast	64	65	t	yes
10	sunny	72	95	f	no
11	sunny	69	70	f	yes
12	rainy	75	80	f	yes
13	sunny	75	70	t	yes
14	overcast	72	90	t	yes
15	overcast	81	75	f	yes
16	rainy	71	91	t	no
17					

Temperature is a numeric value; therefore, you can see min, max, means, and standard deviation in ‘Selected Attribute’ window.
Missing = 0 means that the attribute is specified for all instances (no missing values),
Distinct = 12 means that Temperature has twelve different values, and
Unique = 10 means that other attributes or instances have the same 10 value as Temperature has.
Temperature is a Numeric value; therefore, you can see the statistics describing the distribution of values in the data - Minimum, Maximum, Mean and Standard Deviation. Minimum = 64 is the lowest temperature, Maximum = 85 is the highest temperature, mean and standard deviation. Compare the result with the attribute table “weather.xls”; the numbers in WEKA match the numbers in the table.

You can select a class in the ‘Class’ pull-down box. The last attribute in the ‘Attributes’ window is the default class selected in the ‘Class’ pull-down box.



Filter: None

Current relation
Relation: weather
Instances: 14 Attributes: 5

Selected attribute
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

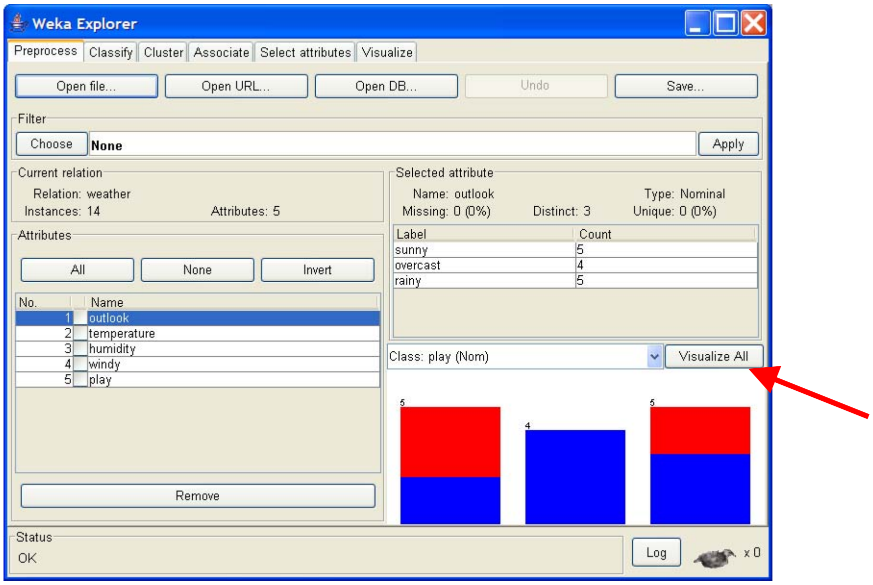
Attributes

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: play (Nom)

Status
OK

You can Visualize the attributes based on selected class. One way is to visualize selected attribute based on class selected in the ‘Class’ pull-down window, or visualize all attributes by clicking on ‘Visualize All’ button.



Filter: None

Relation: weather
Instances: 14 Attributes: 5

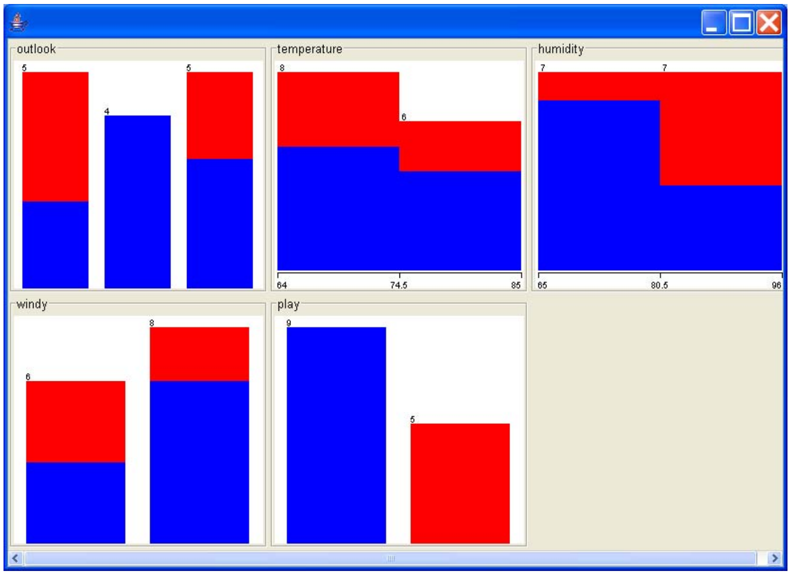
No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

Class: play (Nom)

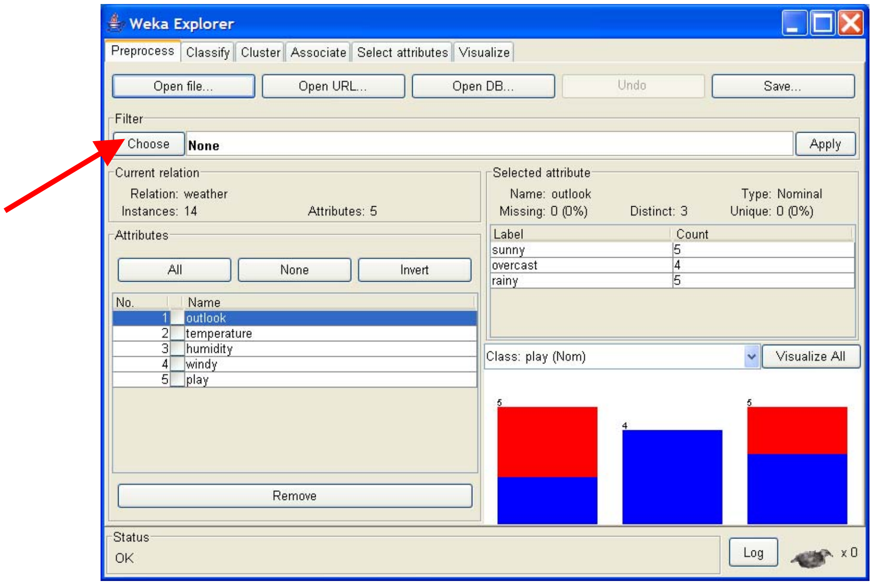
OK



3.6. Setting Filters

Pre-processing tools in WEKA are called “filters”. WEKA contains filters for discretization, normalization, resampling, attribute selection, transformation and combination of attributes [4]. Some techniques, such as association rule mining, can only be performed on categorical data. This requires performing discretization on numeric or continuous attributes [5]. For classification example you do not need to transform the data. For you practice, suppose you need to perform a test on categorical data. There are two attributes that need to be converted: ‘temperature’ and ‘humidity’. In other words, you will keep all of the values for these attributes in the data. This means you can discretize by removing the keyword "numeric" as the type for the ‘temperature’ attribute and replace it with the set of “nominal” values. You can do this by applying a filter.

In ‘Filters’ window, click on the ‘Choose’ button.



Relation: weather
Instances: 14 Attributes: 5

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

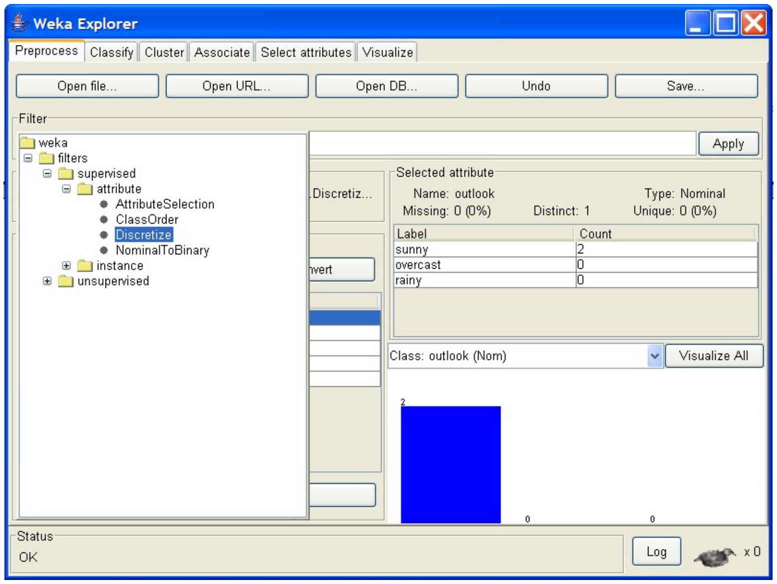
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

Class: play (Nom)

OK

This will show pull-down menu with a list of available filters. Select Supervised → Attribute → Discretize and click on ‘Apply’ button. The filter will convert Numeric values into Nominal.



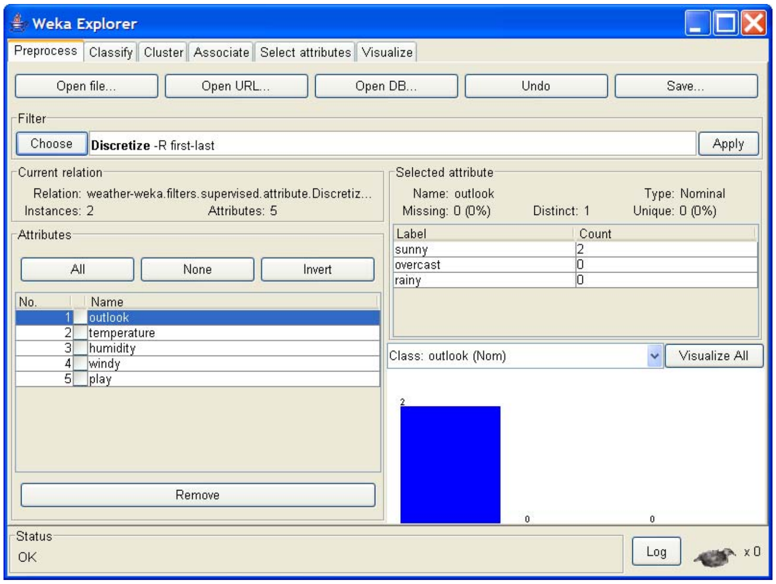
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 1 Unique: 0 (0%)

Label	Count
sunny	2
overcast	0
rainy	0

Class: outlook (Nom)

OK

When filter is chosen, the fields in the window changes to reflect available options.



Filter: **Discretize** -R first-last

Current relation
Relation: weather-weka.filters.supervised.attribute.Discretiz...
Instances: 2 Attributes: 5

Selected attribute
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 1 Unique: 0 (0%)

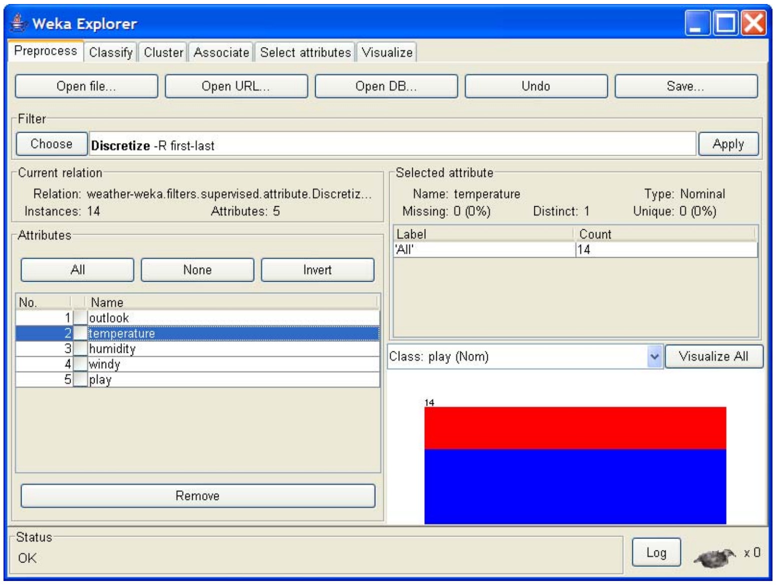
Label	Count
sunny	2
overcast	0
rainy	0

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: outlook (Nom)

Status
OK

As you can see, there is no change in the value Outlook. Select value Temperature, look at the ‘Selected attribute’ box, the ‘Type’ field shows that the attribute type has changed from Numeric to Nominal. The list has changed as well: instead of statistical values there is count of instances, and the count of it is 14 that means that there are 14 instances of the value Temperature.



Filter: **Discretize** -R first-last

Current relation
Relation: weather-weka.filters.supervised.attribute.Discretiz...
Instances: 14 Attributes: 5

Selected attribute
Name: temperature Type: Nominal
Missing: 0 (0%) Distinct: 1 Unique: 0 (0%)

Label	Count
'All'	14

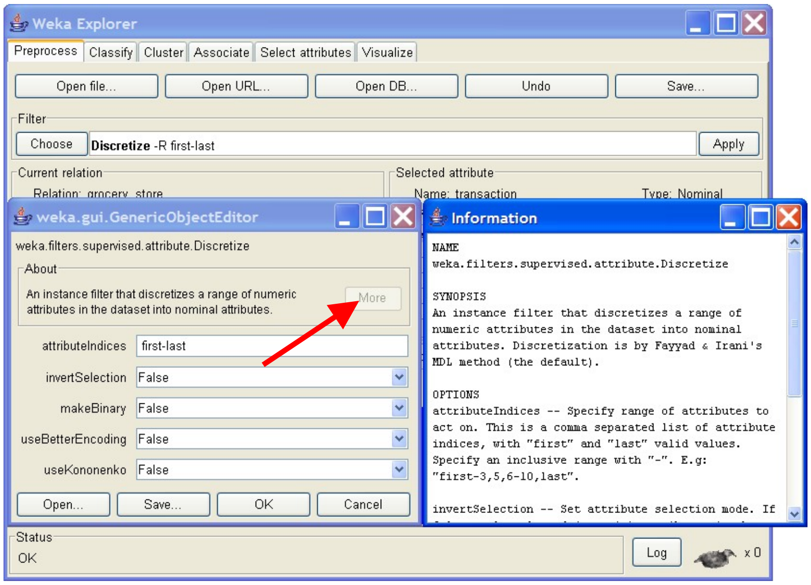
No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Class: play (Nom)

Status
OK

Note, when you right-click on filter, a ‘GenericObjectEditor’ dialog box comes up on your screen. The box lets you to choose the filter configuration options. The same box can be used for classifiers, clusterers and association rules.

Clicking on ‘More’ button brings up an ‘Information’ window describing what the different options can do.



Filter: Discretize -R first-last

Current relation — Relation: grocery_store

Selected attribute — Name: transaction Type: Nominal

weka.filters.supervised.attribute.Discretize

About: An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices	first-last
invertSelection	False
makeBinary	False
useBetterEncoding	False
useKononenko	False

```
NAME
weka.filters.supervised.attribute.Discretize

SYNOPSIS
An instance filter that discretizes a range of
numeric attributes in the dataset into nominal
attributes. Discretization is by Fayyad & Irani's
MDL method (the default).

OPTIONS
attributeIndices -- Specify range of attributes to
act on. This is a comma separated list of attribute
indices, with "first" and "last" valid values.
Specify an inclusive range with "-". E.g:
"first-3,5,6-10,last".

invertSelection -- Set attribute selection mode. If
```

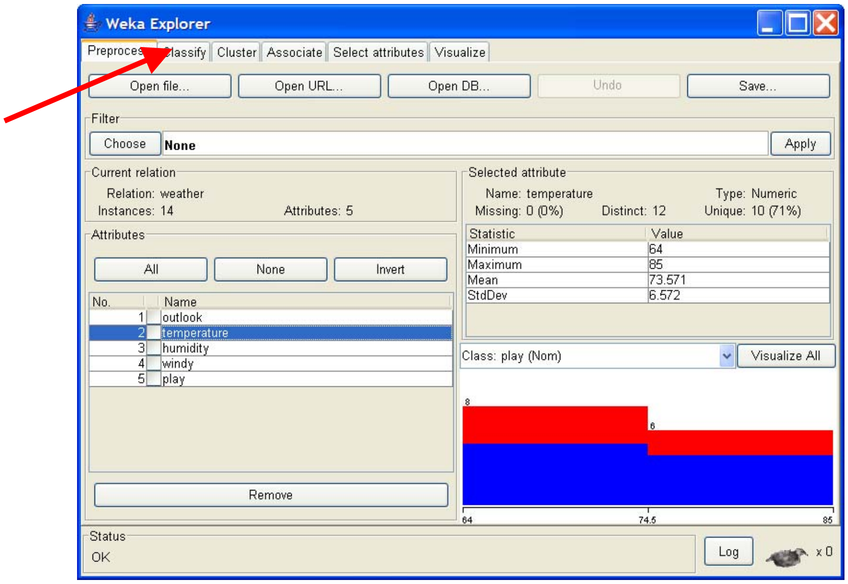
Status: OK

At the bottom of the editor window there are four buttons. ‘Open’ and ‘Save’ buttons allow you to save object configurations for future use. ‘Cancel’ button allows you to exit without saving changes. Once you have made changes, click ‘OK’ to apply them.

4. Building “Classifiers”

Classifiers in WEKA are the models for predicting nominal or numeric quantities. The learning schemes available in WEKA include decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, and bayes’ nets. “Meta”-classifiers include bagging, boosting, stacking, error-correcting output codes, and locally weighted learning [4].

Once you have your data set loaded, all the tabs are available to you. Click on the ‘Classify’ tab.



Filter: None

Current relation — Relation: weather; Instances: 14; Attributes: 5

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

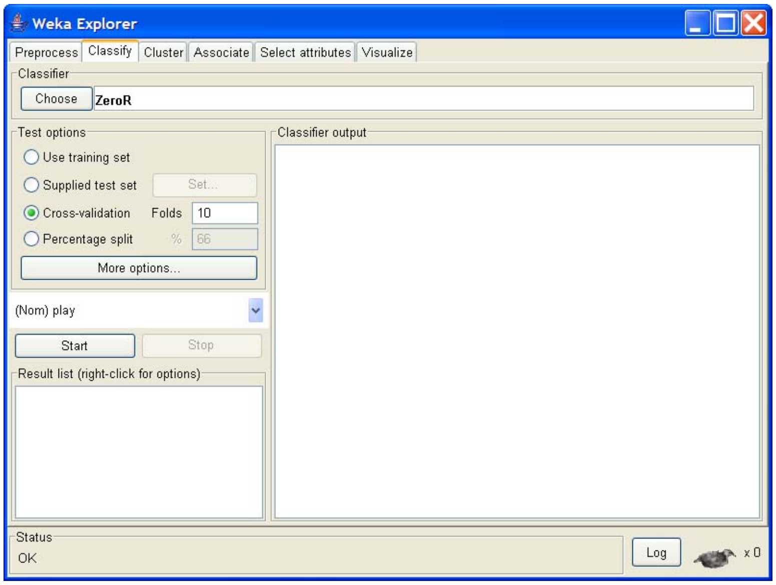
Selected attribute — Name: temperature Type: Numeric; Missing: 0 (0%) Distinct: 12 Unique: 10 (71%)

Statistic	Value
Minimum	64
Maximum	85
Mean	73.571
StdDev	6.572

Class: play (Nom)

Status: OK

‘Classify’ window comes up on the screen.



Classifier: ZeroR

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

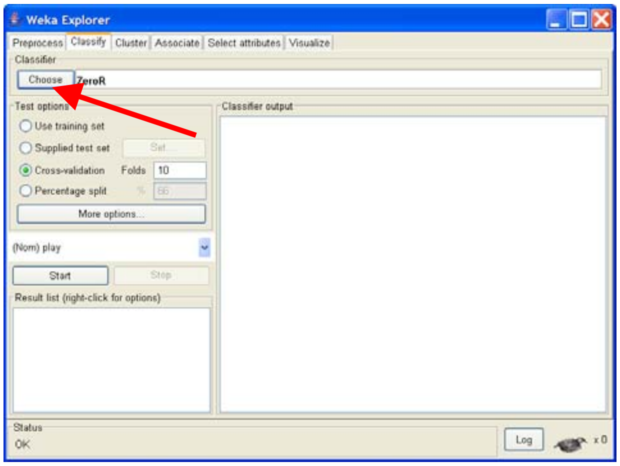
(Nom) play

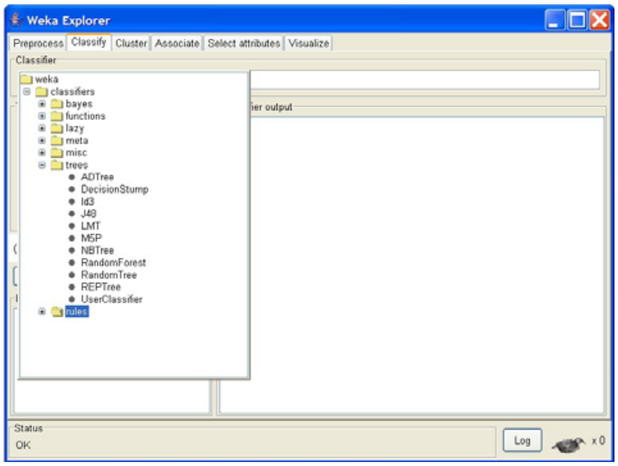
Status: OK

Now you can start analyzing the data using the provided algorithms. In this exercise you will analyze the data with C4.5 algorithm using J48, WEKA's implementation of decision tree learner. The sample data used in this exercise is the weather data from the file "weather.arff". Since C4.5 algorithm can handle numeric attributes, in contrast to the ID3 algorithm from which C4.5 has evolved, there is no need to discretize any of the attributes. Before you start this exercise, make sure you do not have filters set in the 'Preprocess' window. Filter exercise in section 3.6 was just a practice.

4.1. Choosing a Classifier

Click on 'Choose' button in the 'Classifier' box just below the tabs and select C4.5 classifier WEKA → Classifiers → Trees → J48.



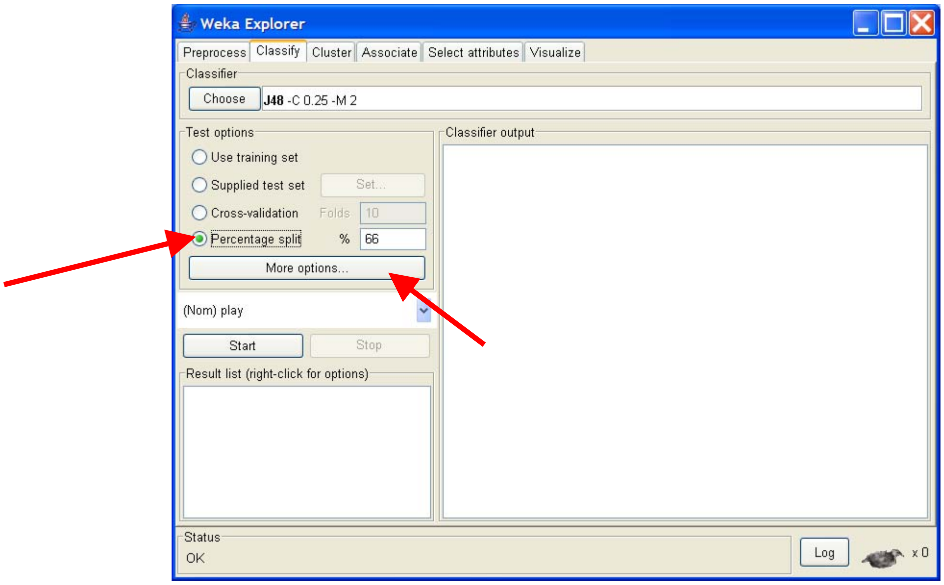


4.2. Setting Test Options

Before you run the classification algorithm, you need to set test options. Set test options in the 'Test options' box. The test options that available to you are [2]:

1. **Use training set.** Evaluates the classifier on haw well it predicts the class of the instances it was trained on.
2. **Supplied test set.** Evaluates the classifier on how well it predicts the class of a set of instances loaded from a file. Clicking on the ‘Set…’ button brings up a dialog allowing you to choose the file to test on.
3. **Cross-validation.** Evaluates the classifier by cross-validation, using the number of folds that are entered in the ‘Folds’ text field.
4. **Percentage split.** Evaluates the classifier on how well it predicts a certain percentage of the data, which is held out for testing. The amount of data held out depends on the value entered in the ‘%’ field.

In this exercise you will evaluate classifier based on how well it predicts 66% of the tested data. Check ‘Percentage split’ radio-button and keep it as default 66%. Click on ‘More options…’ button.



J48 -C 0.25 -M 2

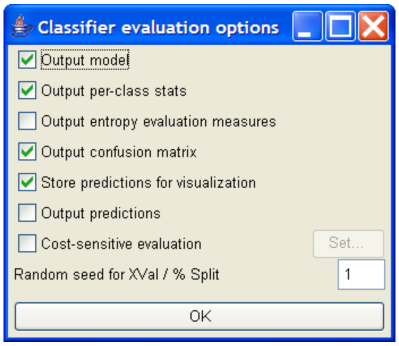
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

OK

Identify what is included into the output. In the ‘Classifier evaluation options’ make sure that the following options are checked [2]:

1. **Output model**. The output is the classification model on the full training set, so that it can be viewed, visualized, etc.
2. **Output per-class stats**. The precision/recall and true/false statistics for each class output.
3. **Output confusion matrix**. The confusion matrix of the classifier’s predictions is included in the output.
4. **Store predictions for visualization**. The classifier’s predictions are remembered so that they can be visualized.
5. Set ‘**Random seed for Xval / % Split**’ to 1. This specifies the random seed used when randomizing the data before it is divided up for evaluation purposes.



Classifier evaluation options

☑ Output model

☑ Output per-class stats

□ Output entropy evaluation measures

☑ Output confusion matrix

☑ Store predictions for visualization

□ Output predictions

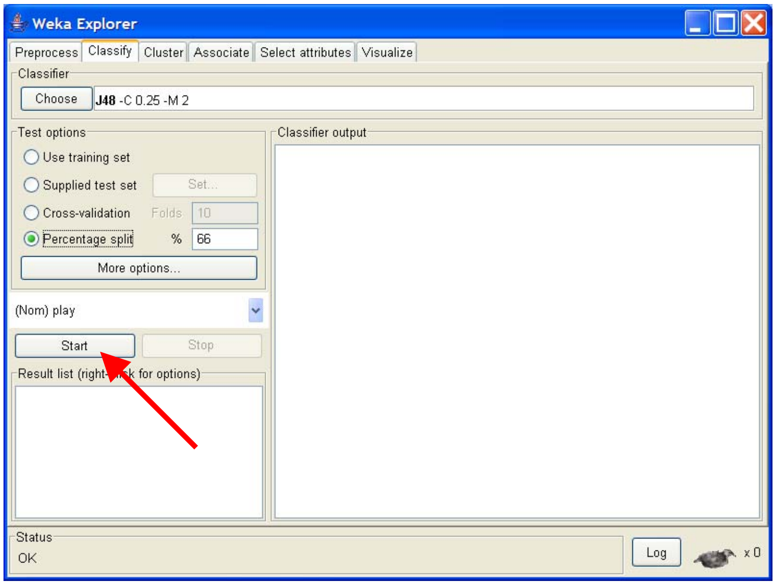
□ Cost-sensitive evaluation

Random seed for XVal / % Split: 1

The remaining options that you do not use in this exercise but that available to you are:

6. **Output entropy evaluation measures**. Entropy evaluation measures are included in the output.
7. **Output predictions**. The classifier's predictions are remembered so that they can be visualized.

Once the options have been specified, you can run the classification algorithm. Click on 'Start' button to start the learning process. You can stop learning process at any time by clicking on 'Stop' button.



Weka Explorer

Classifier: J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

○ Cross-validation Folds 10

● Percentage split % 66

(Nom) play

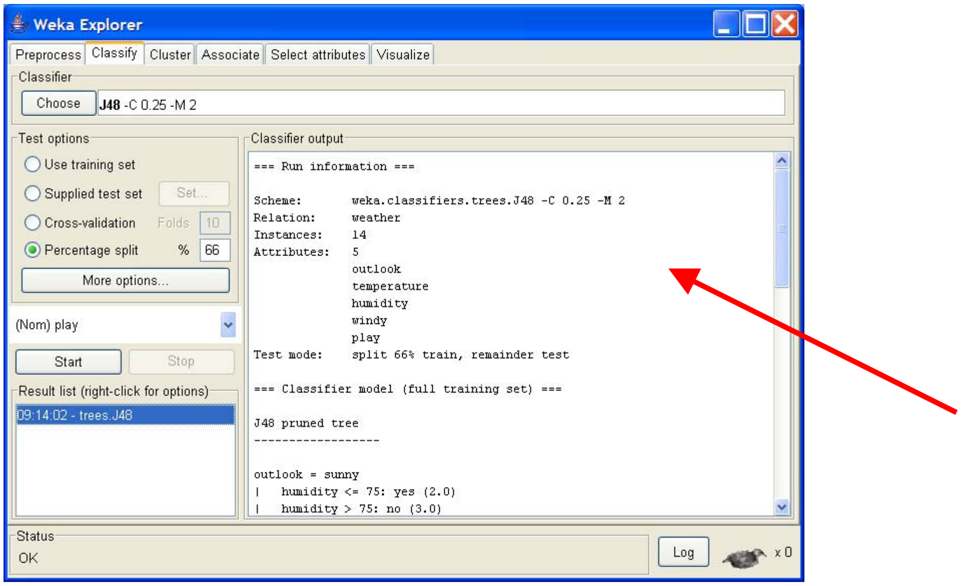
Result list (right-click for options)

Classifier output

Status

OK

When training set is complete, the 'Classifier' output area on the right panel of 'Classify' window is filled with text describing the results of training and testing. A new entry appears in the 'Result list' box on the left panel of 'Classify' window.



Classifier

J48 -C 0.25 -M 2

Test options

○ Use training set

○ Supplied test set

○ Cross-validation Folds 10

● Percentage split % 66

(Nom) play

Result list (right-click for options)

09:14:02 - trees.J48

Classifier output

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    split 66% train, remainder test

=== Classifier model (full training set) ===

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

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
```

Status

OK

4.3. Analyzing Results

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    split 66% train, remainder test

=== Classifier model (full training set) ===

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

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
outlook = overcast: yes (4.0)
outlook = rainy
|   windy = t: no (2.0)
|   windy = f: yes (3.0)

Number of Leaves  :          5

Size of the tree :           8


Time taken to build model: 0.06 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          2          40     %
Incorrectly Classified Instances        3          60     %
Kappa statistic                        -0.3636
Mean absolute error                     0.6
Root mean squared error                 0.7746
Relative absolute error               126.9231 %
Root relative squared error           157.6801 %
Total Number of Instances               5

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall   F-Measure   Class
  0.667     1         0.5         0.667    0.571       yes
  0         0.333     0           0        0            no

=== Confusion Matrix ===

 a b   <-- classified as
 2 1 | a = yes
 2 0 | b = no
```

Run Information gives you the following information:

- the algorithm you used - J48
- the relation name – "weather"
- number of instances in the relation – 14
- number of attributes in the relation – 5 and the list of the attributes: outlook, temperature, humidity, windy, play
- the test mode you selected: split=66%

Classifier model is a pruned decision tree in textual form that was produced on the full training data. As you can see, the first split is on the 'outlook' attribute, at the second level, the splits are on 'humidity' and 'windy'.
In the tree structure, a colon represents the class label that has been assigned to a particular leaf, followed by the number of instances that reach that leaf.
Below the tree structure, there is a number of leaves (which is 5), and the number of nodes in the tree - size of the tree (which is 8).
The program gives a time it took to build the model, which is 0.06 seconds.

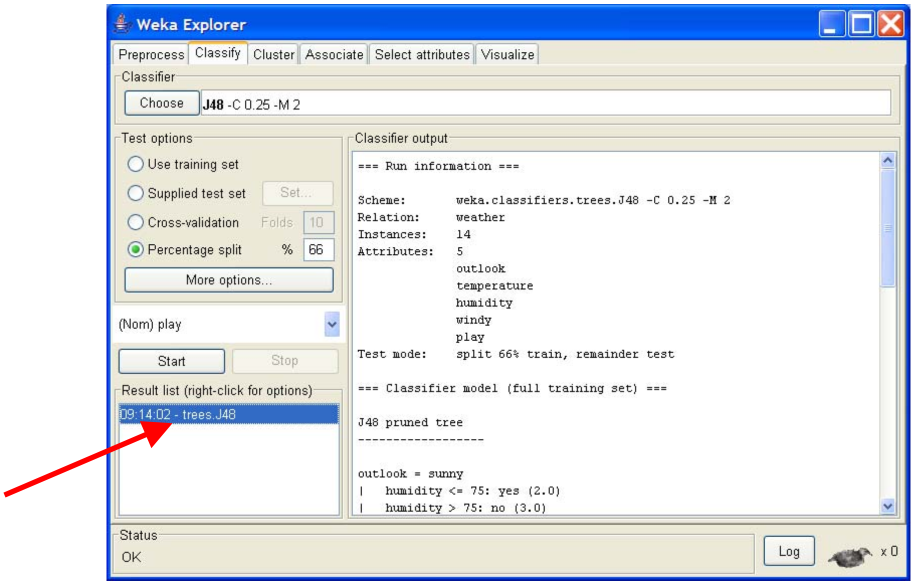
Evaluation on test split. This part of the output gives estimates of the tree's predictive performance, generated by WEKA's evaluation module. It outputs the list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test module. The set of measurements is derived from the training data.
In this case only 40% of 14 training instances have been classified correctly. This indicates that the results obtained from the training data are not optimistic compared with what might be obtained from the independent test set from the same source. In addition to classification error, the evaluation output measurements derived from the class probabilities assigned by the tree. More specifically, it outputs mean output error (0.6) of the probability estimates, the root mean squared error (0.77) is the square root of the quadratic loss. The mean absolute error calculated in a similar way by using the absolute instead of squared difference. The reason that the errors are not 1 or 0 is because not all training instances are classified correctly.

Detailed Accuracy By Class demonstrates a more detailed per-class break down of the classifier's prediction accuracy.

From the Confusion matrix you can see that one instance of a class 'yes' have been assigned to a class 'no', and two of class 'no' are assigned to class 'yes'.

4.4. Visualization of Results

After training a classifier, the result list adds an entry.



Classifier: J48 -C 0.25 -M 2

Test options:
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

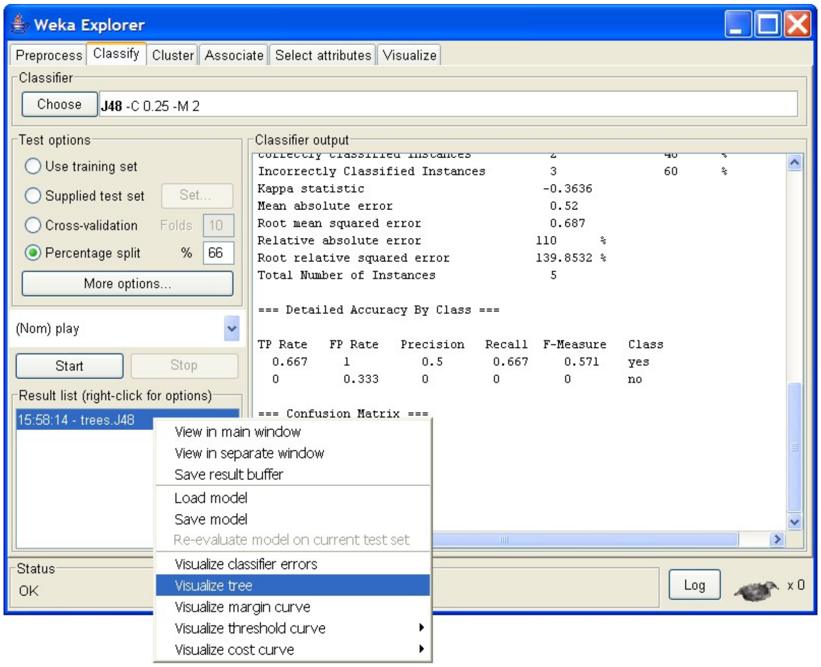
Result list (right-click for options): 09:14:02 - trees.J48

Classifier output:

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Test mode:    split 66% train, remainder test

=== Classifier model (full training set) ===

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

outlook = sunny
|   humidity <= 75: yes (2.0)
|   humidity > 75: no (3.0)
```

Status: OK

WEKA lets you to see a graphical representation of the classification tree. Right-click on the entry in ‘Result list’ for which you would like to visualize a tree. It invokes a menu containing the following items:



Classifier: J48 -C 0.25 -M 2

Test options:
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

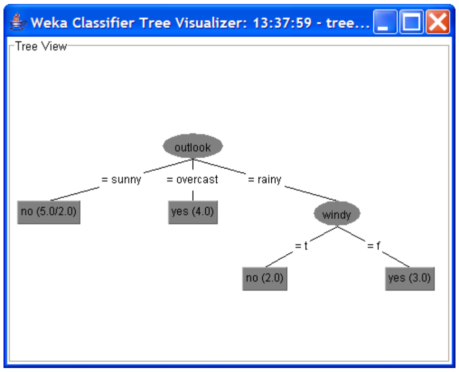
Result list (right-click for options): 15:58:14 - trees.J48

Classifier output:

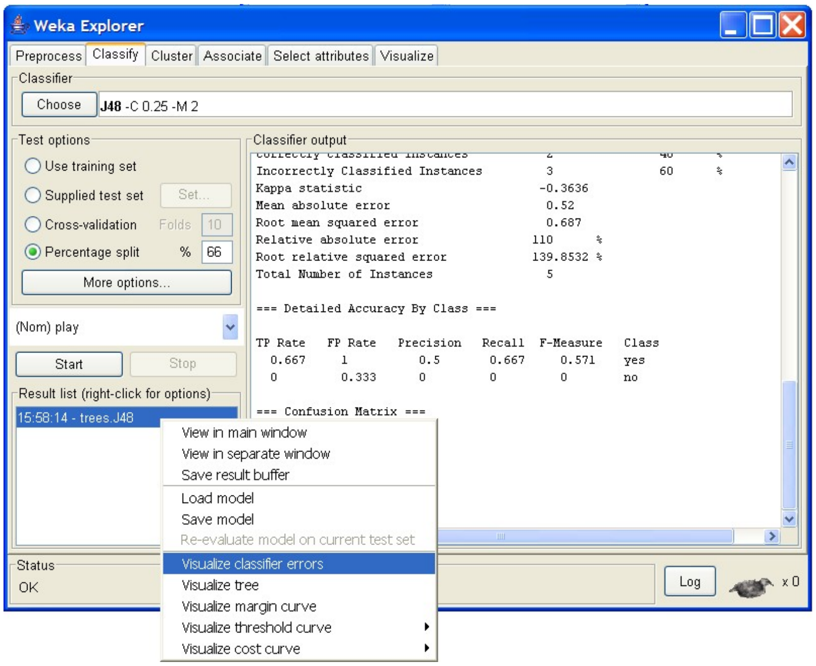
```
Correctly Classified Instances          2               40      %
Incorrectly Classified Instances        3               60      %
Kappa statistic                        -0.3636
Mean absolute error                     0.52
Root mean squared error                 0.687
Relative absolute error               110      %
Root relative squared error           139.8532 %
Total Number of Instances               5

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.667     1          0.5       0.667     0.571    yes
  0         0.333      0         0         0        no

=== Confusion Matrix ===
```

Status: OK

Select the item ‘Visualize tree’; a new window comes up to the screen displaying the tree.



WEKA also lets you to visualize classification errors. Right-click on the entry in ‘Result list’ again and select ‘Visualize classifier errors’ from the menu:



Classifier: J48 -C 0.25 -M 2

○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

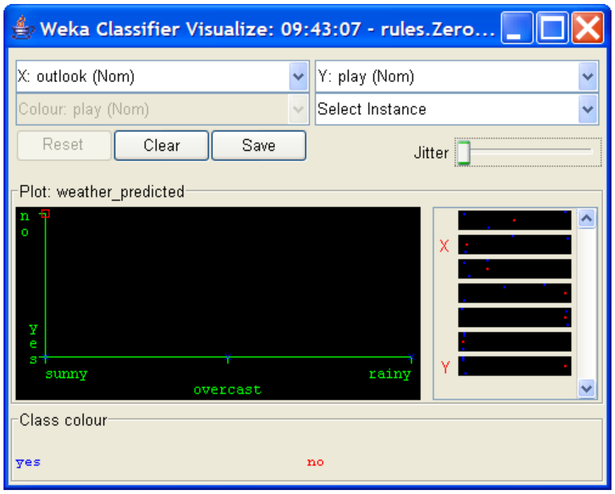
15:58:14 - trees.J48

Status: OK

Classifier output

```
Correctly Classified Instances          2               40      %
Incorrectly Classified Instances        3               60      %
Kappa statistic                        -0.3636
Mean absolute error                     0.52
Root mean squared error                 0.687
Relative absolute error               110      %
Root relative squared error           139.8532 %
Total Number of Instances               5

=== Detailed Accuracy By Class ===

TP Rate   FP Rate   Precision   Recall  F-Measure   Class
  0.667     1          0.5        0.667     0.571    yes
  0         0.333      0          0         0        no

=== Confusion Matrix ===
```

‘Weka Classifier Visualize’ window displaying graph appears on the screen.



X: outlook (Nom)

Y: play (Nom)

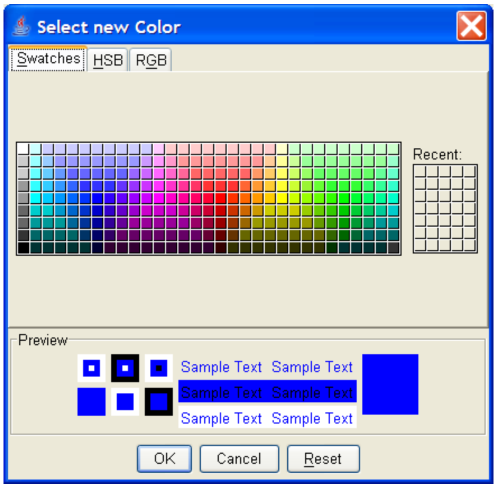
Colour: play (Nom)

Select Instance

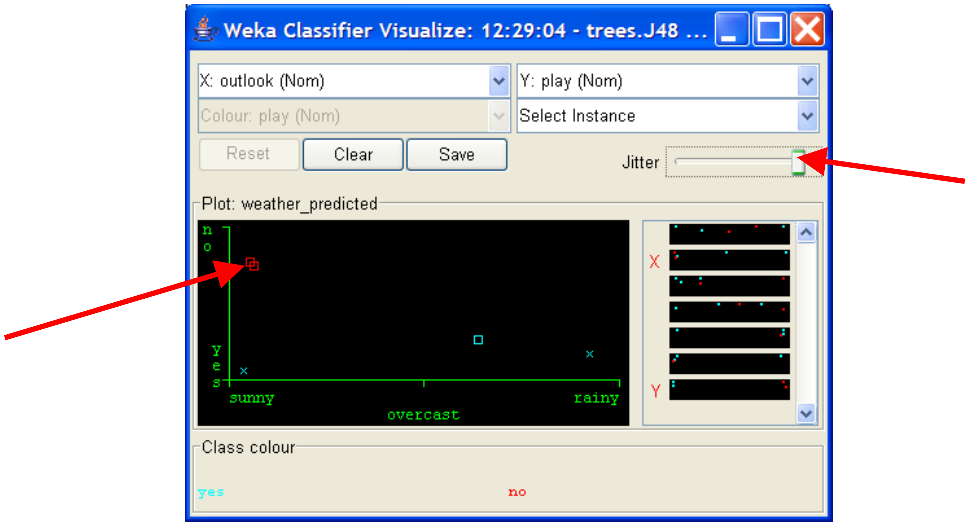
Plot: weather_predicted

Class colour: yes no

On the ‘Weka Classifier Visualize’ window, beneath the X-axis selector there is a drop-down list, ‘Colour’, for choosing the color scheme. This allows you to choose the color of points based on the attribute selected. Below the plot area, there is a legend that describes what values the colors correspond to. In your example, red represents ‘no’, while blue represents ‘yes’. For better visibility you should change the color of label ‘yes’. Left-click on ‘yes’ in the ‘Class colour’ box and select lighter color from the color palette.



To the right of the plot area there are series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution values of the attribute. You can choose what axes are used in the main graph by clicking on these strips (left-click changes X-axis, right-click changes Y-axis).
Change X - axis to ‘Outlook’ attribute and Y - axis to ‘Play’. The instances are spread out in the plot area and concentration points are not visible. Keep sliding ‘Jitter’, a random displacement given to all points in the plot, to the right, until you can spot concentration points.


Weka Classifier Visualize: 12:29:04 - trees.J48 ...

X: outlook (Nom)

Y: play (Nom)

Colour: play (Nom)

Select Instance

Plot: weather_predicted

On the plot you can see the results of classification. Correctly classified instances are represented as crosses, incorrectly classified once represented as squares. In this example in the left lower corner you can see blue cross indicating correctly classified instance: if Outlook = ‘sunny’ → play = ‘yes’.

Look to the upper left corner of the graph, there are two red squares in this corner. The square represents incorrectly classified instance. The following is not correct: if Outlook = ‘sunny’ → play = ‘no’.

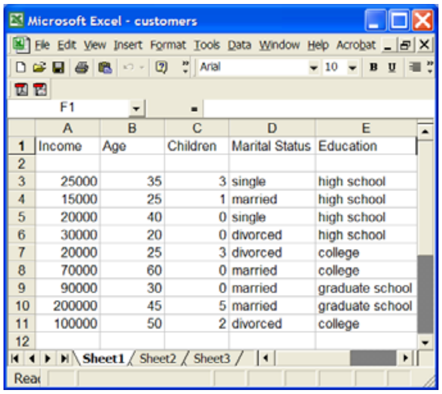
<u>Classification Exercise</u>

Use ID3 algorithm to classify weather data from the “weather.arff” file. Perform initial preprocessing and create a version of the initial dataset in which all numeric attributes should be converted to categorical data.

5. Clustering Data

WEKA contains “clusterers” for finding groups of similar instances in a dataset. The clustering schemes available in WEKA are *k*-Means, EM, Cobweb, *X*-means, FarthestFirst. Clusters can be visualized and compared to “true” clusters (if given). Evaluation is based on log likelihood if clustering scheme produces a probability distribution [4].

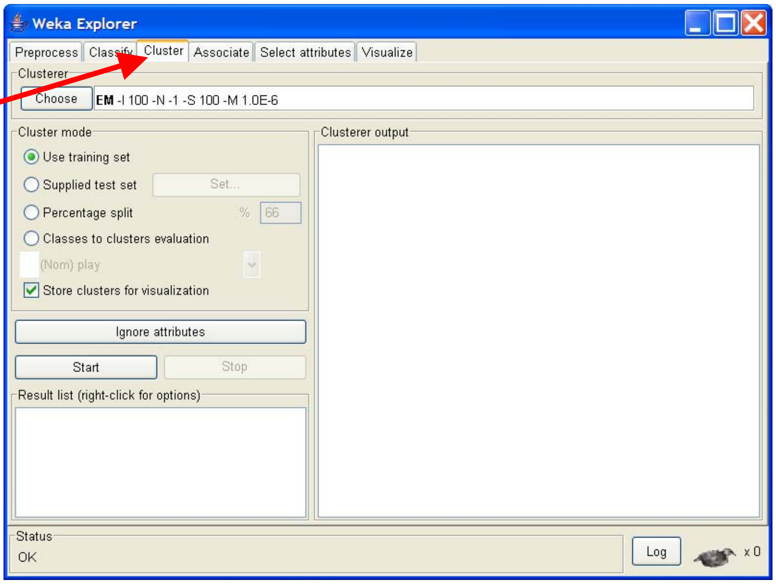
For this exercise we will use customer data [6] that is contained in “customers.arff” file and analyze it with k-means clustering scheme.


Microsoft Excel - customers
File Edit View Insert Format Tools Data Window Help Acrobat
Arial
10
F1
Sheet1
Sheet2
Sheet3
Rea

	A	B	C	D	E
1	Income	Age	Children	Marital Status	Education
2					
3	25000	35	3	single	high school
4	15000	25	1	married	high school
5	20000	40	0	single	high school
6	30000	20	0	divorced	high school
7	20000	25	3	divorced	college
8	70000	60	0	married	college
9	90000	30	0	married	graduate school
10	200000	45	5	married	graduate school
11	100000	50	2	divorced	college
12					

An international online catalog company wishes to group its customers based on common features. Company management does not have any predefined labels for these groups. Based on the outcome of the grouping, they will target marketing and advertising campaigns to the different groups. The information they have about the customers includes income, age, number of children, marital status, and education. For our exercise we will use a part of the database for customers in US. Depending on the type of advertising, not all attributes are important. For example, suppose the advertising is for a special sale on children’s clothes. We will target the advertising only to the persons with young children. The clustering that you will perform in this exercise is as follows. The first group of people has young children and a high school degree, the second group does not have children but has high school degree. The third group has both children and a college degree. The fourth group has higher income and at least a college degree. The fifth group has children and higher degree. Different clustering would have been found by examining either age or marital status.

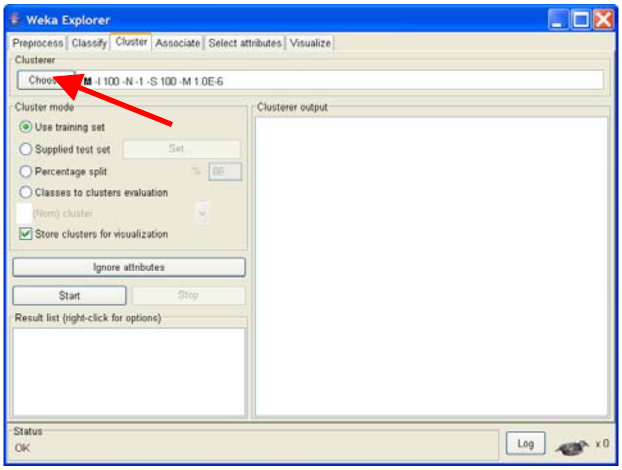
In ‘Preprocess’ window click on ‘Open file…’ button and select “customers.arff” file. Click ‘Cluster’ tab at the top of WEKA Explorer window.



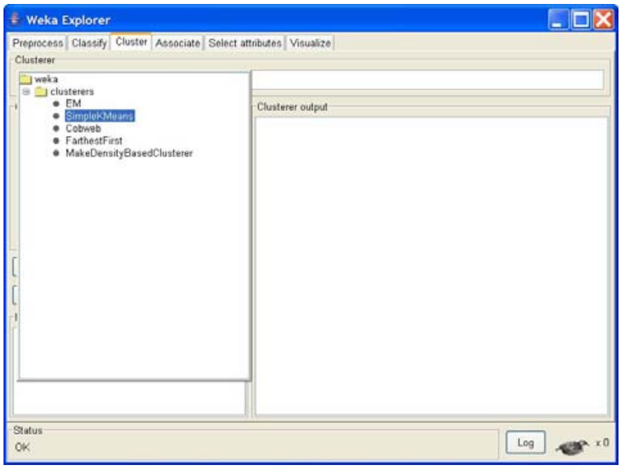
EM -I 100 -N -1 -S 100 -M 1.0E-6

5.1. Choosing Clustering Scheme

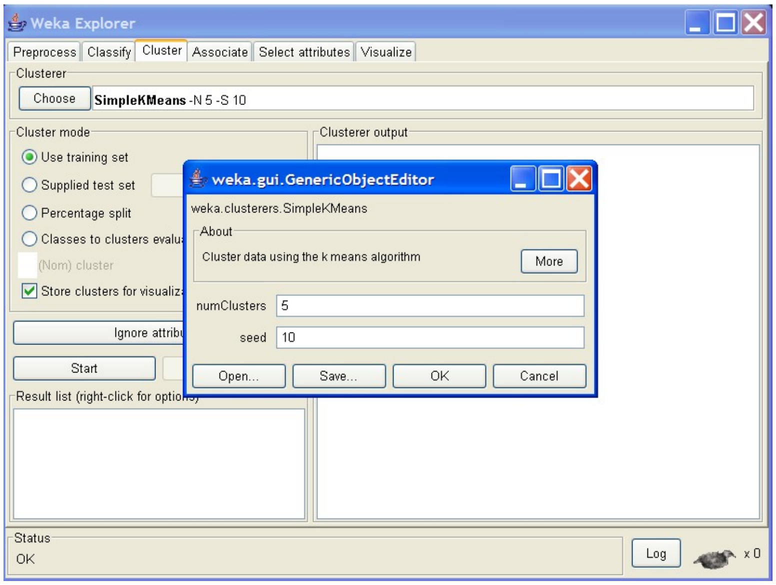
In the ‘Clusterer’ box click on ‘Choose’ button. In pull-down menu select WEKA → Clusterers, and select the cluster scheme ‘SimpleKMeans’. Some implementations of K-means only allow numerical values for attributes; therefore, we do not need to use a filter.



EM -I 100 -N -1 -S 100 -M 1.0E-6



Once the clustering algorithm is chosen, right-click on the algorithm, “weak.gui.GenericObjectEditor” comes up to the screen. Set the value in “numClusters” box to 5 (instead of default 2) because you have five clusters in your .arff file. Leave the value of ‘seed’ as is. The seed value is used in generating a random number, which is used for making the initial assignment of instances to clusters. Note that, in general, K-means is quite sensitive to how clusters are initially assigned. Thus, it is often necessary to try different values and evaluate the results.



SimpleKMeans -N 5 -S 10

weka.clusterers.SimpleKMeans

Cluster data using the k means algorithm

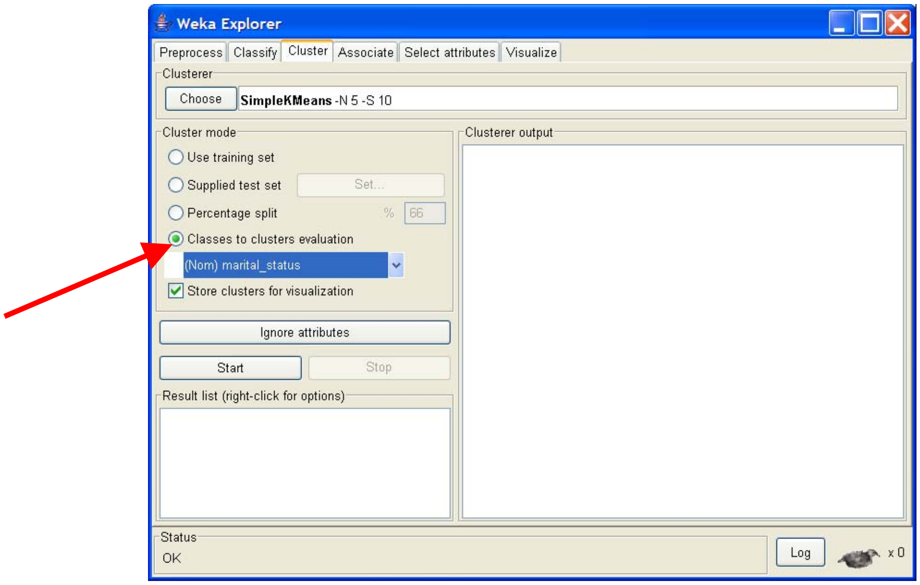
numClusters 5

seed 10

OK

5.2. Setting Test Options

Before you run the clustering algorithm, you need to choose 'Cluster mode'. Click on 'Classes to cluster evaluation' radio-button in 'Cluster mode' box and select 'marital_status' in the pull-down box below. It means that you will compare how well the chosen clusters match up with a pre-assigned class ('marital_status') in the data.

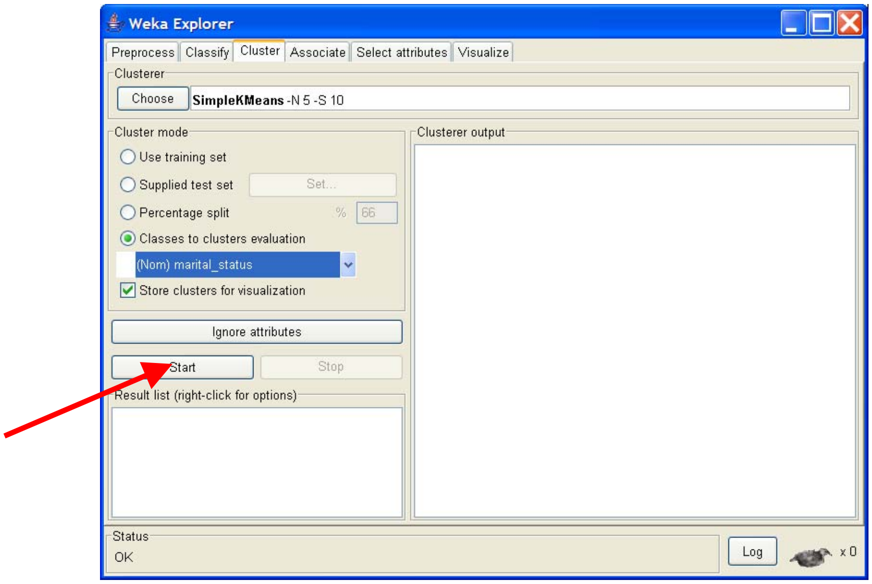


SimpleKMeans -N 5 -S 10

(Nom) marital_status

OK

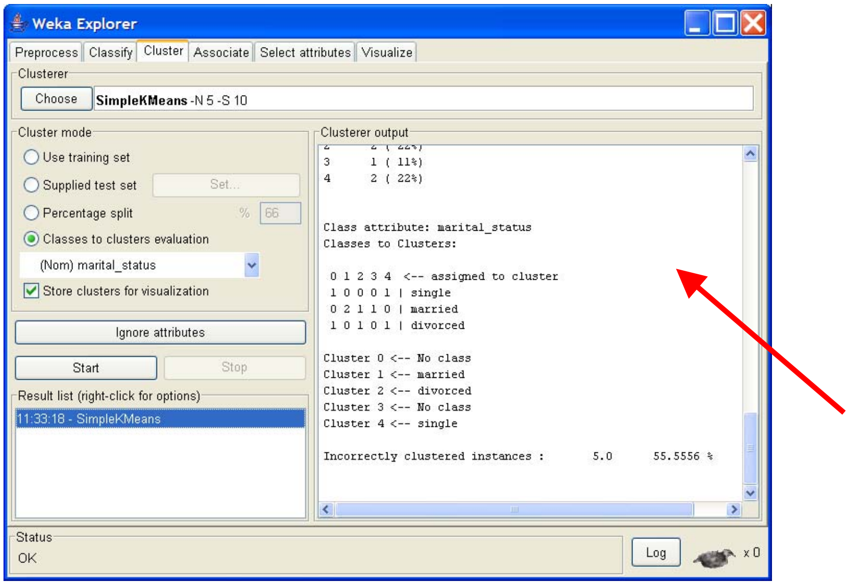
Once the options have been specified, you can run the clustering algorithm. Click on the 'Start' button to execute the algorithm.



SimpleKMeans -N 5 -S 10

○ Use training set
○ Supplied test set Set...
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) marital_status
☑ Store clusters for visualization

When training set is complete, the 'Cluster' output area on the right panel of 'Cluster' window is filled with text describing the results of training and testing. A new entry appears in the 'Result list' box on the left of the result. These behave just like their classification counterparts.



SimpleKMeans -N 5 -S 10

○ Use training set
○ Supplied test set Set...
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) marital_status
☑ Store clusters for visualization

11:33:18 - SimpleKMeans

```
3       1 ( 11%)
4       2 ( 22%)


Class attribute: marital_status
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 1 0 0 0 1 | single
 0 2 1 1 0 | married
 1 0 1 0 1 | divorced

Cluster 0 <-- No class
Cluster 1 <-- married
Cluster 2 <-- divorced
Cluster 3 <-- No class
Cluster 4 <-- single

Incorrectly clustered instances :       5.0      55.5556 %
```

5.3. Analyzing Results

```
=== Run information ===

Scheme:     weka.clusterers.SimpleKMeans -N 5 -S 10
Relation:   customers
Instances:  9
Attributes: 5
            income
            age
            children
            education
Ignored:
            marital_status
Test mode:  Classes to clusters evaluation on training data

=== Clustering model (full training set) ===
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 3.449558299853908

Cluster centroids:

Cluster 0
            Mean/Mode:  22500       30        3       high_school
            Std Devs:   3535.5339   7.0711    N/A     N/A
Cluster 1
            Mean/Mode:  145000      37.5      0       graduate_school
            Std Devs:   77781.7459  10.6066   N/A     N/A
Cluster 2
            Mean/Mode:  85000       55        0       college
            Std Devs:   21213.2034  7.0711    N/A     N/A
Cluster 3
            Mean/Mode:  15000       25        1       high_school
            Std Devs:   0           0         N/A     N/A
Cluster 4
            Mean/Mode:  25000       30        0       high_school
            Std Devs:   7071.0678   14.1421   N/A     N/A

=== Evaluation on training set ===

kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 6.899116599707816

Cluster centroids:

Cluster 0
            Mean/Mode:  22500       30        3       high_school
            Std Devs:   3535.5339   7.0711    N/A     N/A
Cluster 1
            Mean/Mode:  145000      37.5      0       graduate_school
            Std Devs:   77781.7459  10.6066   N/A     N/A
Cluster 2
            Mean/Mode:  85000       55        0       college
            Std Devs:   21213.2034  7.0711    N/A     N/A
Cluster 3
            Mean/Mode:  15000       25        1       high_school
            Std Devs:   0           0         N/A     N/A
Cluster 4
            Mean/Mode:  25000       30        0       high_school
            Std Devs:   7071.0678   14.1421   N/A     N/A

Clustered Instances

0     2 ( 22%)
1     2 ( 22%)
2     2 ( 22%)
3     1 ( 11%)
4     2 ( 22%)

Class attribute: marital_status
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 1 0 0 0 1 | single
 0 2 1 1 0 | married
 1 0 1 0 1 | divorced

Cluster 0 <-- No class
Cluster 1 <-- married
Cluster 2 <-- divorced
Cluster 3 <-- No class
Cluster 4 <-- single

Incorrectly clustered instances :      5.0       55.5556 %
```

'Run Information' gives you the following information:

- the clustering scheme used: SimpleKMeans with 5 clusters
- the relation name "customers"
- number of instances in the relation – 9
- number of attributes in the relation – 6
- list of attributes used in clustering
- the ignored cluster 'marital_status' is an attribute the clustering is performed on.

The clustering model shows the centroid of each cluster and statistics on the number and percentage of instances assigned to different clusters. Cluster centroids are the mean vectors for each cluster; so, each dimension value and the centroid represents the mean value for that dimension in the cluster. Thus, centroids can be used to characterize the clusters.
WEKA generated clusters are:
Cluster 0 shows that this is a segment of cases representing 25 and 35 year old, either single or divorced, people with income $22,500 in average, who have 3 children.
In cluster 1 there are 30 and 45 year old married people who do not have children.
In cluster 2 there are 50 and 60 year old married and divorced people with higher income college degree and no children.
Cluster 3 represents 25 year old married people with one child lower income and high school degree.
Cluster 4 represents 20 and 40 year old single and divorced people with lower income, high school degree and no children.

Sum of errors within the clusters is recalculated.

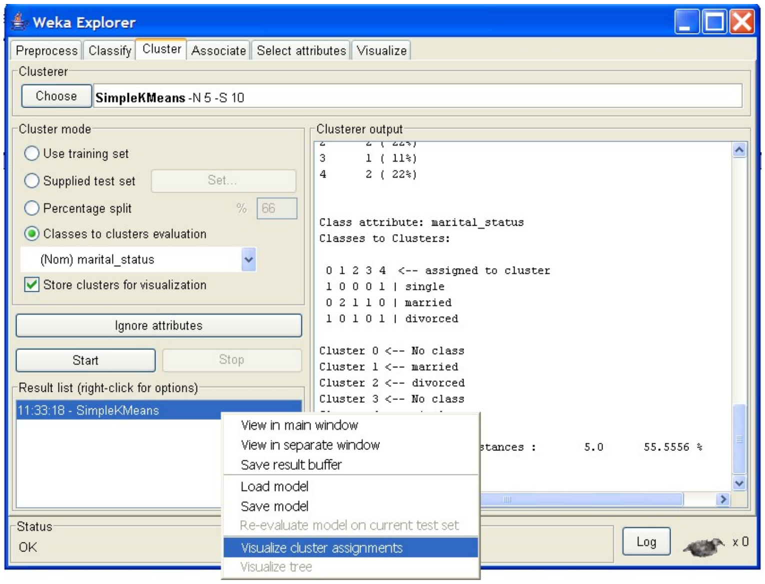
'Cluster Instances' section shows the number of instances in each new cluster.
For example, cluster 3 has 1 instance: people of age 25 who have one child.
Cluster 4 has 2 instances: people of age 30 in average (including 20 and 40 y.o.), whose average income is $25,000, with high school education and no children.

'Classes to Clusters" represents class ('marital-status') assigned to clusters.

The last line displays the you have 5 number incorrectly classified instances, which is 55.5 %.

5.4. Visualization of Results

Another way of representation of results of clustering is through visualization. Right-click on the entry in the ‘Result list’ and select ‘Visualize cluster assignments’ in the pull-down window.



Clusterer: SimpleKMeans -N 5 -S 10

Cluster mode:
- ○ Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ● Classes to clusters evaluation
- (Nom) marital_status
- ☑ Store clusters for visualization

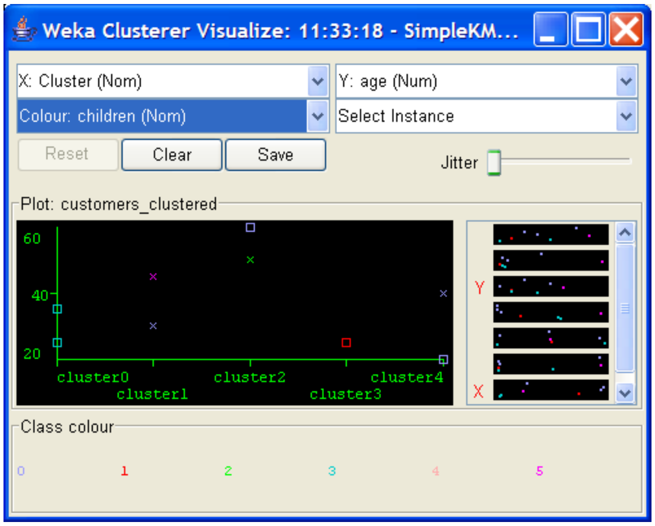
Result list (right-click for options): 11:33:18 - SimpleKMeans

Status: OK

Clusterer output:

```
2       2 ( 22%)
3       1 ( 11%)
4       2 ( 22%)


Class attribute: marital_status
Classes to Clusters:

 0 1 2 3 4  <-- assigned to cluster
 1 0 0 0 1 | single
 0 2 1 1 0 | married
 1 0 1 0 1 | divorced

Cluster 0 <-- No class
Cluster 1 <-- married
Cluster 2 <-- divorced
Cluster 3 <-- No class

stances :       5.0      55.5556 %
```

This brings up the ‘Weka Clusterer Visualize’ window.



X: Cluster (Nom)

Y: age (Num)

Colour: children (Nom)

Select Instance

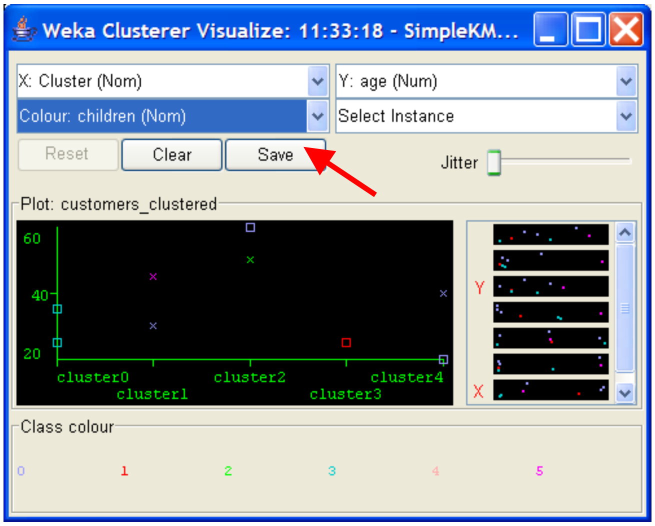
Plot: customers_clustered

On the ‘Weka Clusterer Visualize’ window, beneath the X-axis selector there is a drop-down list, ‘Colour’, for choosing the color scheme. This allows you to choose the color of points based on the attribute selected. Below the plot area, there is a legend that describes what values the colors correspond to. In your example, seven different colors represent seven numbers (number of children). For better visibility you should change the color of label ‘3’. Left-click on ‘3’ in the ‘Class colour’ box and select lighter color from the color palette.

To the right of the plot area there are series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution values of the attribute. You can choose what axes are used in the main graph by clicking on these strips (left-click changes X-axis, right-click changes Y-axis). Set X - axis to ‘Cluster’ attribute, Y - axis to ‘Age’. Select ‘Children’ as the color dimension. You can see the result in a visual rendering of the relationship within each cluster. For instance, you can note that ‘cluster 0’ represents a group of people of age 25 and 35, who have 3 children, ‘cluster 1’ represents a group of people of age 30 and 45 who do not have children, ‘cluster 2’ represents 50 and 60 year old people with no children, ‘cluster 3’ represents 25 year old married people with one child, and ‘cluster 4’ represents 20 and 40 year old people without children.

The initially correctly clustered instances are represented by crosses, incorrectly clustered once represented as squares. By changing the color dimension to other attributes, you can see their distribution within each of the clusters.

You may want to save the resulting data set, which included each instance along with its assigned cluster. To do so, click ‘Save’ button in the visualization window and save the result as the file “customers_kmeans.arff”.



Weka Clusterer Visualize: 11:33:18 - SimpleKM...

X: Cluster (Nom)

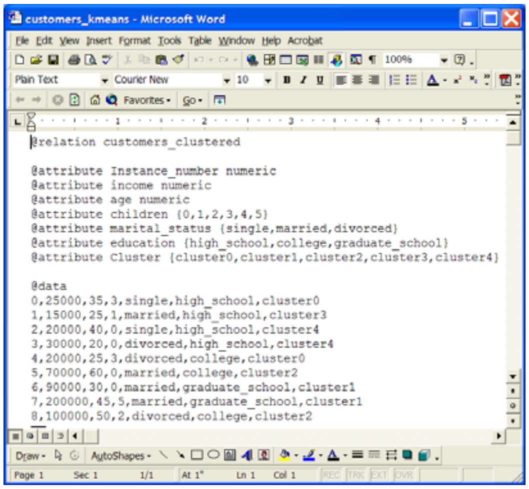
Y: age (Num)

Colour: children (Nom)

Select Instance

Plot: customers_clustered

As you can see, there is a new attribute appeared in the file – ‘cluster’ that was added by WEKA. This attribute represents the custering done by WEKA.



customers_kmeans - Microsoft Word

```
@relation customers_clustered

@attribute Instance_number numeric
@attribute income numeric
@attribute age numeric
@attribute children {0,1,2,3,4,5}
@attribute marital_status {single,married,divorced}
@attribute education {high_school,college,graduate_school}
@attribute Cluster {cluster0,cluster1,cluster2,cluster3,cluster4}

@data
0,25000,35,3,single,high_school,cluster0
1,15000,25,1,married,high_school,cluster3
2,20000,40,0,single,high_school,cluster4
3,30000,20,0,divorced,high_school,cluster4
4,20000,25,3,divorced,college,cluster0
5,70000,60,0,married,college,cluster2
6,90000,30,0,married,graduate_school,cluster1
7,200000,45,5,married,graduate_school,cluster1
8,100000,50,2,divorced,college,cluster2
```

File Edit View Insert Format Tools Table Window Help Acrobat
100%
Plain Text
Courier New
10
Favorites
Go
Draw
AutoShapes
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1 REC TRK EXT OVR

Clustering Exercise

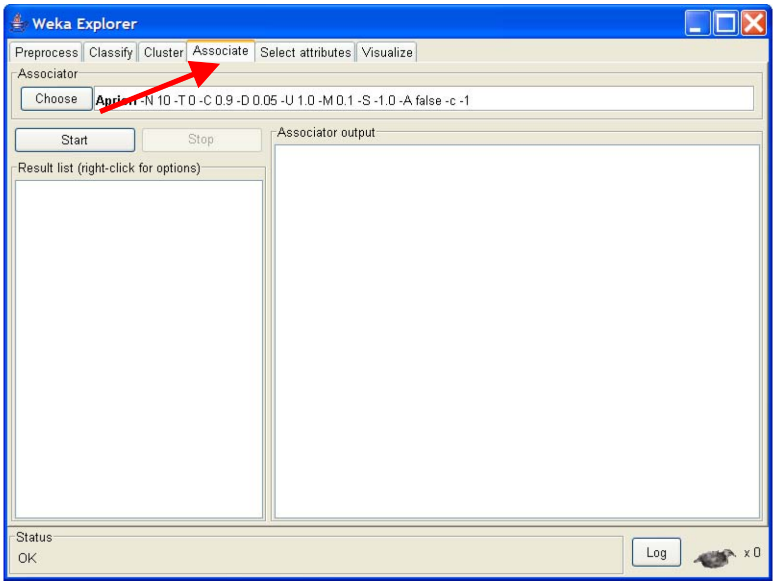
Use k-means algorithm to bank data from the “bank.arff” file. Perform initial preprocessing and create a version of the initial data set in which the ID field should be removed and the "children" attribute should be converted to categorical data.

6. Finding Associations

WEKA contains an implementation of the Apriori algorithm for learning association rules. This is the only currently available scheme for learning associations in WEKA. It works only with discrete data and will identify statistical dependencies between groups of attributes, milk, peanut butter and bread, jelly, beer and diapers, with confidence 40% and support 30%. Apriori can compute all rules that have a given minimum support and exceed a given confidence.

6.1. Choosing Association Scheme

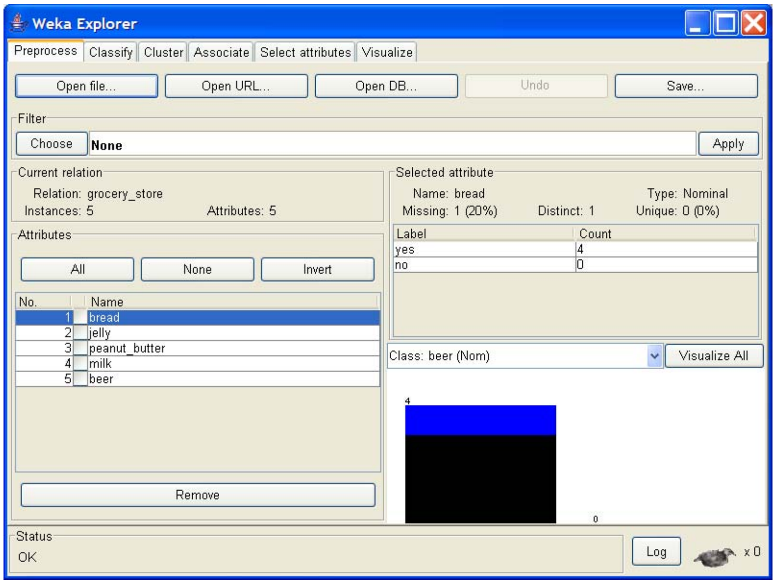
Click ‘Associate’ tab at the top of ‘WEKA Explorer’ window. It brings up interface for the Apriori algorithm.



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A false -c -1

Status: OK

The association rule scheme cannot handle numeric values; therefore, for this exercise you will use grocery store data from the “grocery.arff” file where all values are nominal. Go back to ‘Preprocessing’ section described in part 4 and open “grocery.arff” file.



Filter: None

Relation: grocery_store
Instances: 5 Attributes: 5

Name: bread Type: Nominal
Missing: 1 (20%) Distinct: 1 Unique: 0 (0%)

Label	Count
yes	4
no	0

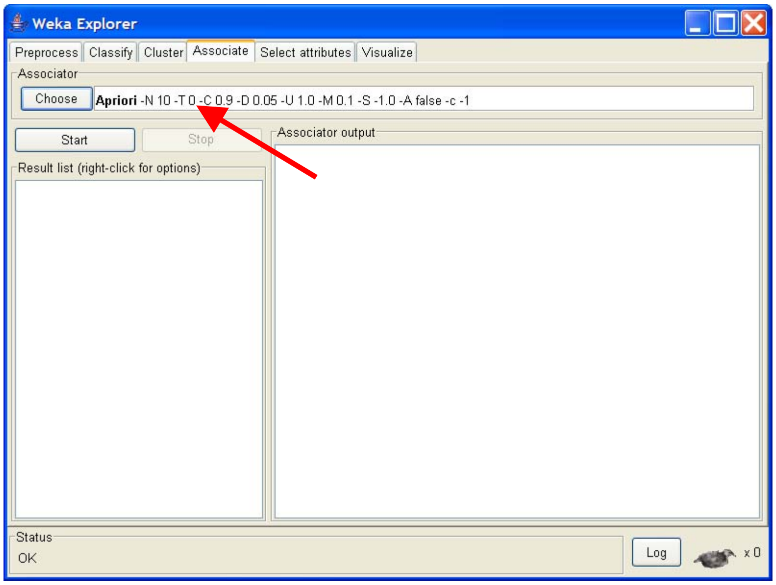
No.	Name
1	bread
2	jelly
3	peanut_butter
4	milk
5	beer

Class: beer (Nom)

Status
OK

6.2. Setting Test Options

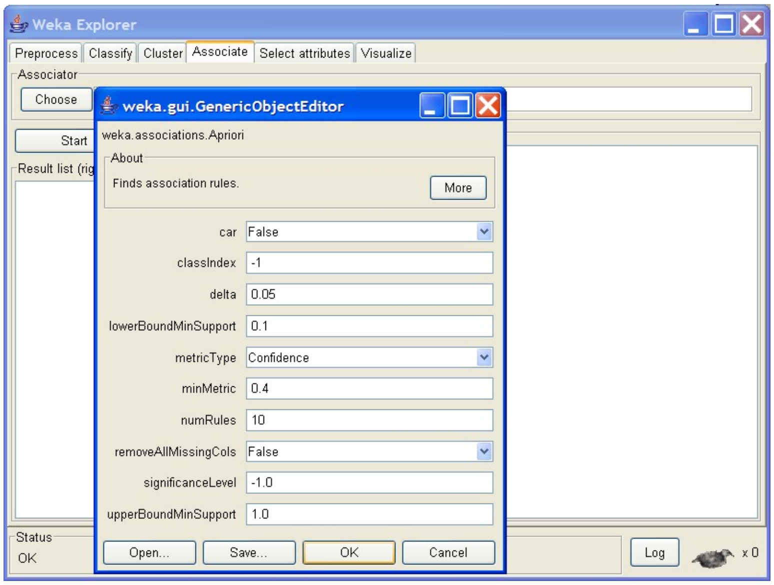
Check the text field in the ‘Associator’ box at the top of the window. As you can see, there are no other associators to choose and no extra options for testing the learning scheme.



Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A false -c -1

Status
OK

Right-click on the ‘Associator’ box, ‘GenericObjectEditor’ appears on your screen. In the dialog box, change the value in ‘minMetric’ to 0.4 for confidence = 40%. Make sure that the default value of rules is set to 100. The upper bound for minimum support ‘upperBoundMinSupport’ should be set to 1.0 (100%) and ‘lowerBoundMinSupport’ to 0.1. Apriori in WEKA starts with the upper bound support and incrementally decreases support (by delta increments, which by default is set to 0.05 or 5%). The algorithm halts when either the specified number of rules is generated, or the lower bound for minimum support is reached. The ‘significanceLevel’ testing option is only applicable in the case of confidence and is (-1.0) by default (not used).



weka.associations.Apriori

About

Finds association rules.

car: False

classIndex: -1

delta: 0.05

lowerBoundMinSupport: 0.1

metricType: Confidence

minMetric: 0.4

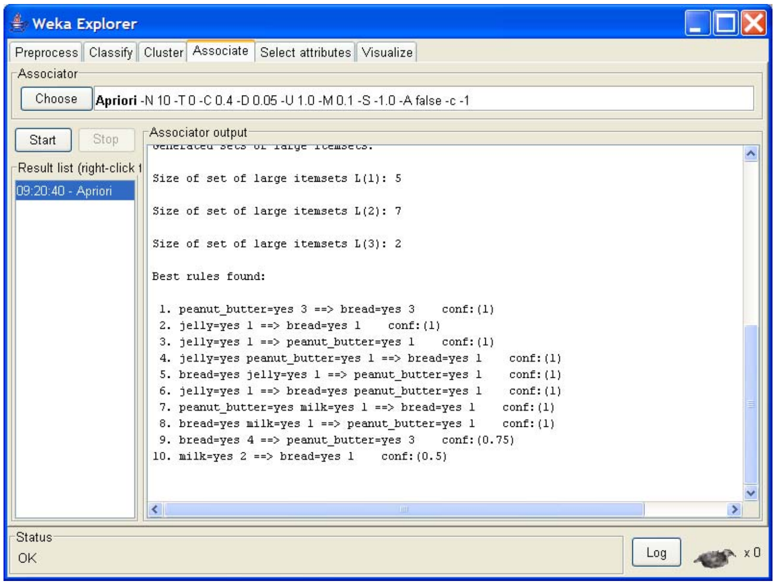
numRules: 10

removeAllMissingCols: False

significanceLevel: -1.0

upperBoundMinSupport: 1.0

Once the options have been specified, you can run Apriori algorithm. Click on the ‘Start’ button to execute the algorithm.



Apriori -N 10 -T 0 -C 0.4 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A false -c -1

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 5

Size of set of large itemsets L(2): 7

Size of set of large itemsets L(3): 2

Best rules found:

 1. peanut_butter=yes 3 ==> bread=yes 3    conf:(1)
 2. jelly=yes 1 ==> bread=yes 1    conf:(1)
 3. jelly=yes 1 ==> peanut_butter=yes 1    conf:(1)
 4. jelly=yes peanut_butter=yes 1 ==> bread=yes 1    conf:(1)
 5. bread=yes jelly=yes 1 ==> peanut_butter=yes 1    conf:(1)
 6. jelly=yes 1 ==> bread=yes peanut_butter=yes 1    conf:(1)
 7. peanut_butter=yes milk=yes 1 ==> bread=yes 1    conf:(1)
 8. bread=yes milk=yes 1 ==> peanut_butter=yes 1    conf:(1)
 9. bread=yes 4 ==> peanut_butter=yes 3    conf:(0.75)
10. milk=yes 2 ==> bread=yes 1    conf:(0.5)
```

6.3. Analyzing Results

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.4 -D 0.05 -U 1.0 -M 0.1 -S -
1.0 -A false -c -1
Relation:     grocery_store
Instances:    5
Attributes:   5
              bread
              jelly
              peanut_butter
              milk
              beer
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.3
Minimum metric <confidence>: 0.4
Number of cycles performed: 14

Generated sets of large itemsets:

Size of set of large itemsets L(1): 5

Size of set of large itemsets L(2): 7

Size of set of large itemsets L(3): 2

Best rules found:

 1. peanut_butter=yes 3 ==> bread=yes 3    conf:(1)
 2. jelly=yes 1 ==> bread=yes 1    conf:(1)
 3. jelly=yes 1 ==> peanut_butter=yes 1    conf:(1)
 4. jelly=yes peanut_butter=yes 1 ==> bread=yes 1    conf:(1)
 5. bread=yes jelly=yes 1 ==> peanut_butter=yes 1    conf:(1)
 6. jelly=yes 1 ==> bread=yes peanut_butter=yes 1    conf:(1)
 7. peanut_butter=yes milk=yes 1 ==> bread=yes 1    conf:(1)
 8. bread=yes milk=yes 1 ==> peanut_butter=yes 1    conf:(1)
 9. bread=yes 4 ==> peanut_butter=yes 3    conf:(0.75)
10. milk=yes 2 ==> bread=yes 1    conf:(0.5)
```

Run Information gives you the following information:

- the scheme for learning association we used - Apriori
- the relation name – “grocery_store”
- number of instances in the relation – 5
- number of attributes in the relation – 4 and the list of attributes

The results for Apriori algorithm are the following:
First, the program generated the sets of large itemsets found for each support size considered. In this case five item sets of three items were found to have the required minimum support.

By default, Apriori tries to generate ten rules. It begins with a minimum support of 100% of the data items and decreases this in steps of 5% until there are at least ten rules with the required minimum confidence, or until the support has reached a lower bound of 10% whichever occurs first. The minimum confidence is set 0.4 (40%). As you can see, the minimum support decreased to 0.3 (30%), before the required number of rules can be generated. Generation of the required number of rules involved a total of 14 iterations.

The last part gives the association rules that are found. The number preceding = => symbol indicates the rule’s support, that is, the number of items covered by its premise. Following the rule is the number of those items for which the rule’s consequent holds as well. In the parentheses there is a confidence of the rule.

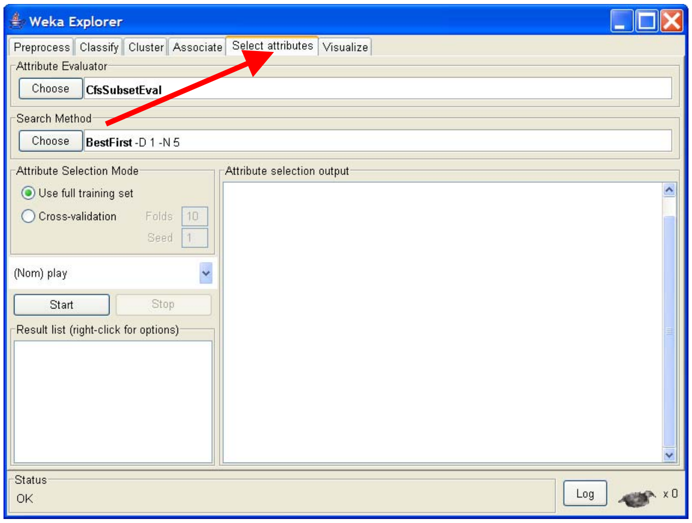
Association Rules Exercise

Use Apriori algorithm to generate association rules for Iris data from the “iris.arff” file. Perform initial preprocessing and create a version of the initial data set in which the numeric attributes should be converted to categorical data.

7. Attribute Selection

Attribute selection searches through all possible combinations of attributes in the data and finds which subset of attributes works best for prediction [1]. Attribute selection methods contain two parts: a search method such as best-first, forward selection, random, exhaustive, genetic algorithm, ranking, and an evaluation method such as correlation-based, wrapper, information gain, chi-squared. Attribute selection mechanism is very flexible - WEKA allows (almost) arbitrary combinations of the two methods [4].

For this exercise you will use weather data from the “weather.arff” file. To begin an attribute selection, click ‘Select attributes’ tab.



Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) play

Status: OK

7.1. Selecting Options

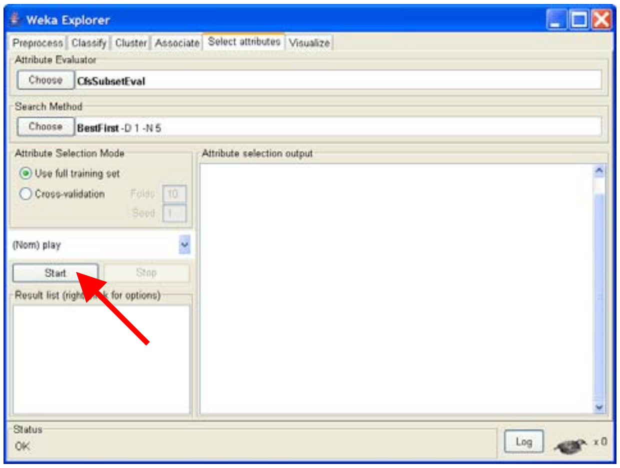
To search through all possible combinations of attributes in the data and find which subset of attributes works best for prediction, make sure that you set up attribute evaluator to ‘CfsSubsetEval’ and a search method to ‘BestFirst’. The evaluator will determine what method to use to assign a worth to each subset of attributes. The search method will determine what style of search to perform.

The options that you can set for selection in the ‘Attribute Selection Mode’ box are [2]:

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.

Specify which attribute to treat as the class in the drop-down box below the test options.

Once all the test options are set, you can start the attribute selection process by clicking on ‘Start’ button.



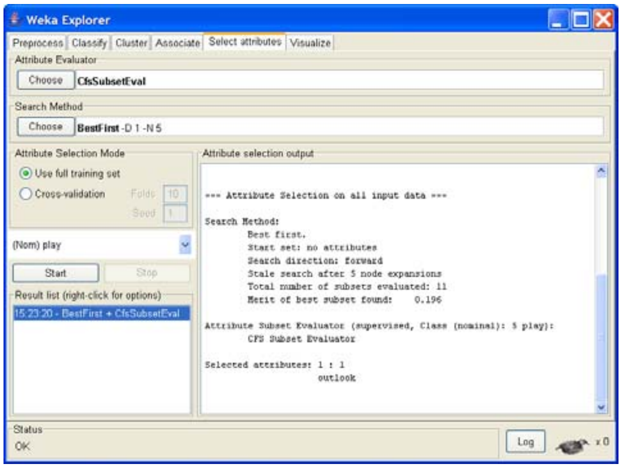
Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) play

Status: OK



Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

(Nom) play

15:23:20 - BestFirst + CfsSubsetEval

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

Search Method:
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 11
        Merit of best subset found:    0.196

Attribute Subset Evaluator (supervised, Class (nominal): 5 play):
        CFS Subset Evaluator

Selected attributes: 1 : 1
                     outlook
```

Status: OK

When it is finished, the results of selection are shown on the right part of the window and entry is added to the ‘Result list’.

7.2. Analyzing Results

```
 === Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:       weka.attributeSelection.BestFirst -D 1 -N 5
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
              play
Evaluation mode:    evaluate on all training data



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

Search Method:
            Best first.
            Start set: no attributes
            Search direction: forward
            Stale search after 5 node expansions
            Total number of subsets evaluated: 11
            Merit of best subset found:    0.196

Attribute Subset Evaluator (supervised, Class (nominal): 5
play):
            CFS Subset Evaluator

Selected attributes: 1 : 1
                      outlook
```

Run Information gives you the following information:

- the evaluator we used – CfsSubsetEval
- the search method - BestFit
- the relation name – “weather”
- number of instances in the relation – 14
- number of attributes in the relation – 5 and the list of attributes

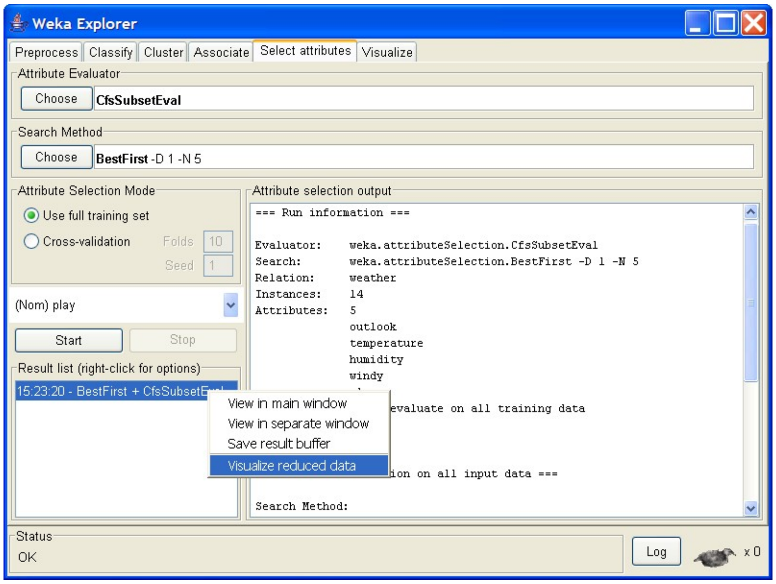
The search method selected is the Best Fit. The software started search with no attributes, and it is forward search. We evaluated 11 subsets and the merit of the best subset is 0.196.

The attribute evaluator used is CFS Subset Evaluator. We used supervised learning with labels in the attribute ‘play’.

The selected attribute for prediction is ‘outlook’.

7.3. Visualizing Results

Right-click on the entry in the ‘Result list’. From the pull-down menu select ‘Visualize reduced data’.



Attribute Evaluator: CfsSubsetEval

Search Method: BestFirst -D 1 -N 5

Attribute Selection Mode
● Use full training set
○ Cross-validation Folds 10 Seed 1

(Nom) play

Result list (right-click for options)
15:23:20 - BestFirst + CfsSubsetEval

Attribute selection output

```
=== Run information ===

Evaluator:    weka.attributeSelection.CfsSubsetEval
Search:       weka.attributeSelection.BestFirst -D 1 -N 5
Relation:     weather
Instances:    14
Attributes:   5
              outlook
              temperature
              humidity
              windy
                              evaluate on all training data



                              ion on all input data ===

Search Method:
```

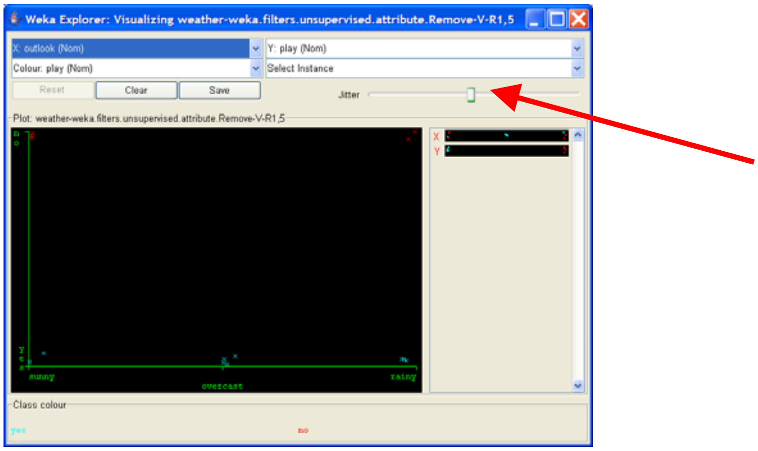
Status
OK

In the window below you can see a prediction for ‘play’ depending on the ‘outlook’. For better visibility the color of label ‘yes’ was changed to the lighter one and ‘Jitter’ was slid to the right to see concentration points.

In the WEKA visualization window, beneath the X-axis selector there is a drop-down list, ‘Colour’, for choosing the color scheme. This allows you to choose the color of points based on the attribute selected. Below the plot area, there is a legend that describes what values the colors correspond to. In your example, red represents ‘no’, while blue represents ‘yes’. For better visibility you should change the color of label ‘yes’. Left-click on ‘yes’ in the ‘Class colour’ box and select lighter color from the color palette.

To the right of the plot area there are series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution values of the attribute. You can choose what axes are used in the main graph by clicking on these strips (left-click changes X-axis, right-click changes Y-axis).

Change X - axis to ‘Outlook’ attribute and Y - axis to ‘Play’. The instances are spread out in the plot area and concentration points are not visible. Keep sliding ‘Jitter’, a random displacement given to all points in the plot, to the right, until you can spot concentration points.


Weka Explorer: Visualizing weather-weka.filters.unsupervised.attribute.Remove-V-R1,5
Reset
Clear
Save
Jitter
no
yes
sunny
overcast
rainy
X
Y
yes
no

X: outlook (Nom)

Y: play (Nom)

Colour: play (Nom)

Select Instance

Plot: weather-weka.filters.unsupervised.attribute.Remove-V-R1,5

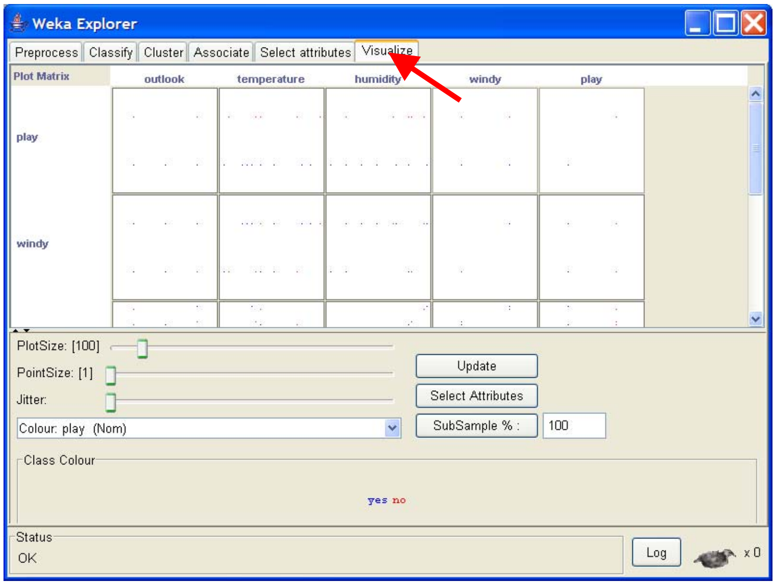
Class colour

The prediction is as follows: if the ‘outlook’ is sunny, play = ‘yes’, and if the ‘outlook’ is ‘rainy’, play = ‘no’, which is very likely to happen. There are few instances displayed in the window that may or may not happen: if ‘outlook’ = ‘sunny’, ‘play’ = ‘no’ and if ‘outlook’ = ‘rainy’, ‘play’ = ‘yes’. Note, in this section there are no correcty or incorrectly classified symbols in the graph because the result is based on probability.

8. Data Visualization

WEKA’s visualization allows you to visualize a 2-D plot of the current working relation. Visualization is very useful in practice, it helps to determine difficulty of the learning problem. WEKA can visualize single attributes (1-d) and pairs of attributes (2-d), rotate 3-d visualizations (Xgobi-style). WEKA has “Jitter” option to deal with nominal attributes and to detect “hidden” data points [4].

To open Visualization screen, click ‘Visualize’ tab.



PlotSize: [100]

PointSize: [1]

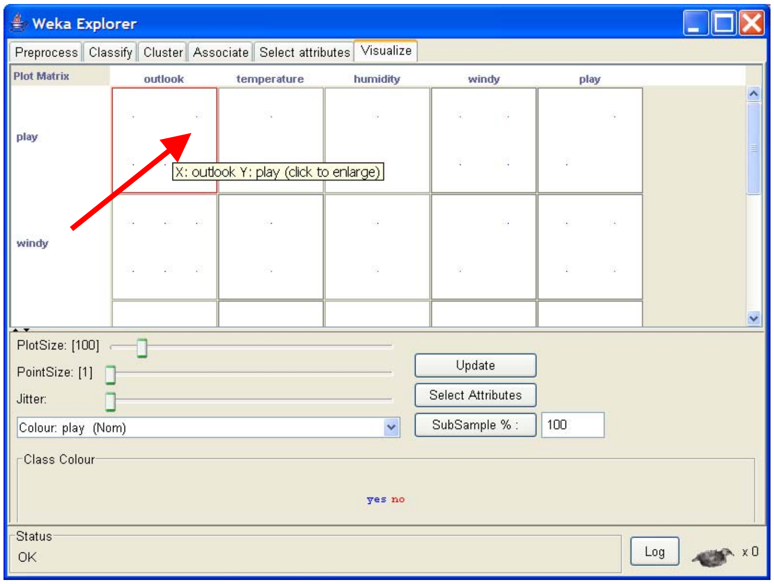
Jitter:

Colour: play (Nom)

SubSample % : 100

OK

Select a square that corresponds to the attributes you would like to visualize. For example, let’s choose ‘outlook’ for X – axis and ‘play’ for Y – axis. Click anywhere inside the square that corresponds to ‘play on the left and ‘outlook’ at the top.



PlotSize: [100]

PointSize: [1]

Jitter:

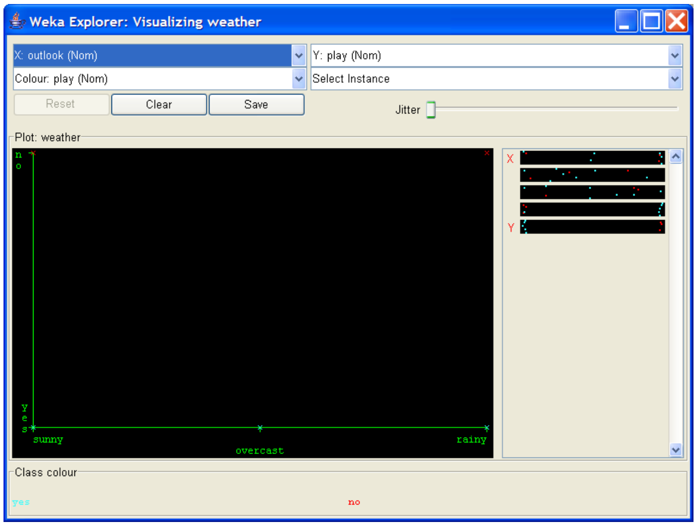
Colour: play (Nom)

SubSample % : 100

Status

OK

A ‘Visualizing weather’ window appears on the screen.



X: outlook (Nom)

Y: play (Nom)

Colour: play (Nom)

Jitter

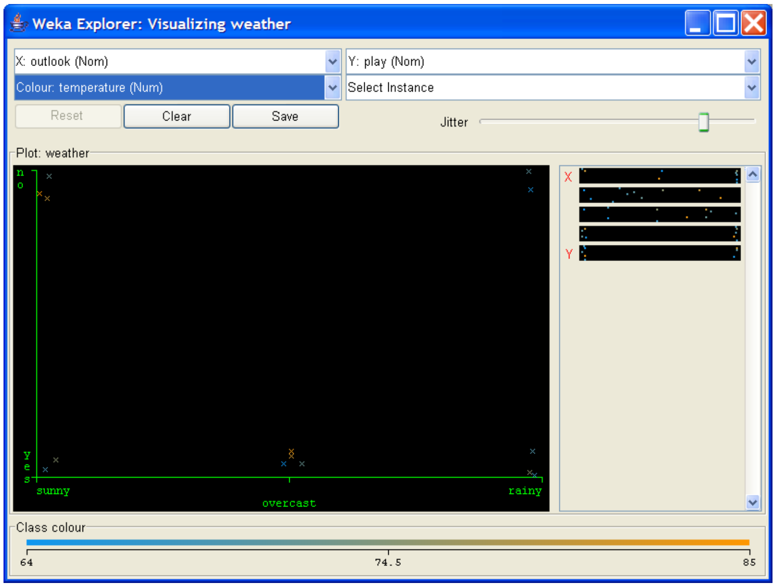
8.1. Changing the View

In the visualization window, beneath the X-axis selector there is a drop-down list, ‘Colour’, for choosing the color scheme. This allows you to choose the color of points based on the attribute selected. Below the plot area, there is a legend that describes what values the colors correspond to. In your example, red represents ‘no’, while blue represents ‘yes’. For better visibility you should change the color of label ‘yes’. Left-click on ‘yes’ in the ‘Class colour’ box and select lighter color from the color palette.

To the right of the plot area there are series of horizontal strips. Each strip represents an attribute, and the dots within it show the distribution values of the attribute. You can choose what axes are used in the main graph by clicking on these strips (left-click changes X-axis, right-click changes Y-axis).

The software sets X - axis to ‘Outlook’ attribute and Y - axis to ‘Play’. The instances are spread out in the plot area and concentration points are not visible. Keep sliding ‘Jitter’, a random displacement given to all points in the plot, to the right, until you can spot concentration points.

The results are shown below. But on this screen we changed ‘Colour’ to temperature. Besides ‘outlook’ and ‘play’, this allows you to see the ‘temperature’ corresponding to the ‘outlook’. It will affect your result because if you see ‘outlook’ = ‘sunny’ and ‘play’ = ‘no’ to explain the result, you need to see the ‘temperature’ – if it is too hot, you do not want to play. Change ‘Colour’ to ‘windy’, you can see that if it is windy, you do not want to play as well.



Weka Explorer: Visualizing weather

X: outlook (Nom)

Y: play (Nom)

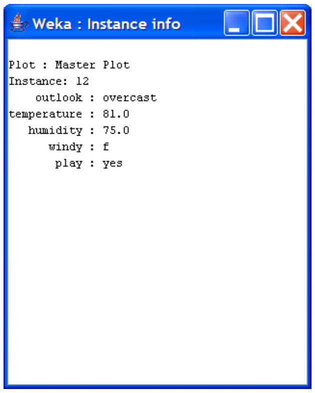
Colour: temperature (Num)

Select Instance

8.2. Selecting Instances

Sometimes it is helpful to select a subset of the data using visualization tool. A special case is the ‘UserClassifier’, which lets you to build your own classifier by interactively selecting instances. Below the Y – axis there is a drop-down list that allows you to choose a selection method. A group of points on the graph can be selected in four ways [2]:

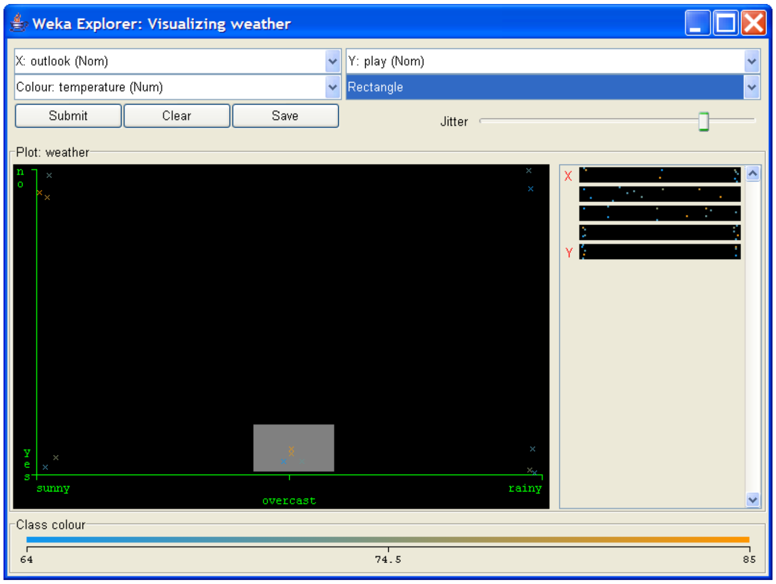
1. **Select Instance**. Click on an individual data point. It brings up a window listing attributes of the point. If more than one point will appear at the same location, more than one set of attributes will be shown.



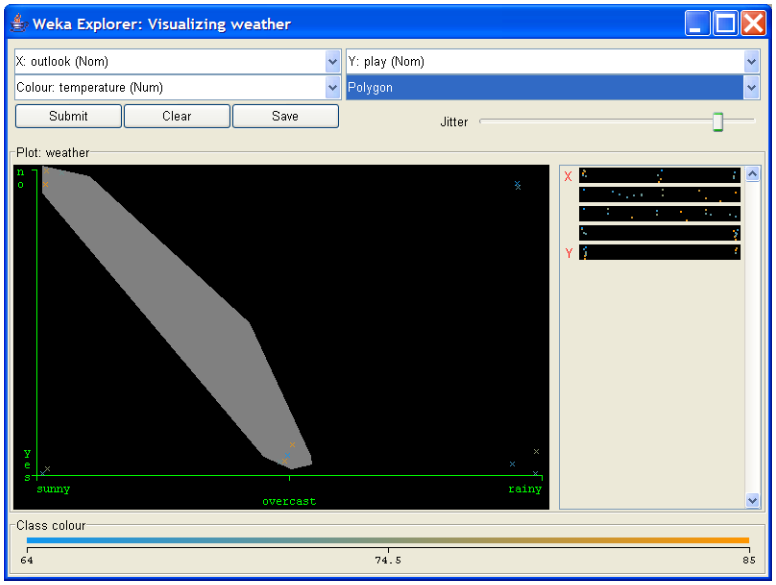
Weka : Instance info

```
Plot : Master Plot
Instance: 12
    outlook : overcast
temperature : 81.0
   humidity : 75.0
      windy : f
       play : yes
```

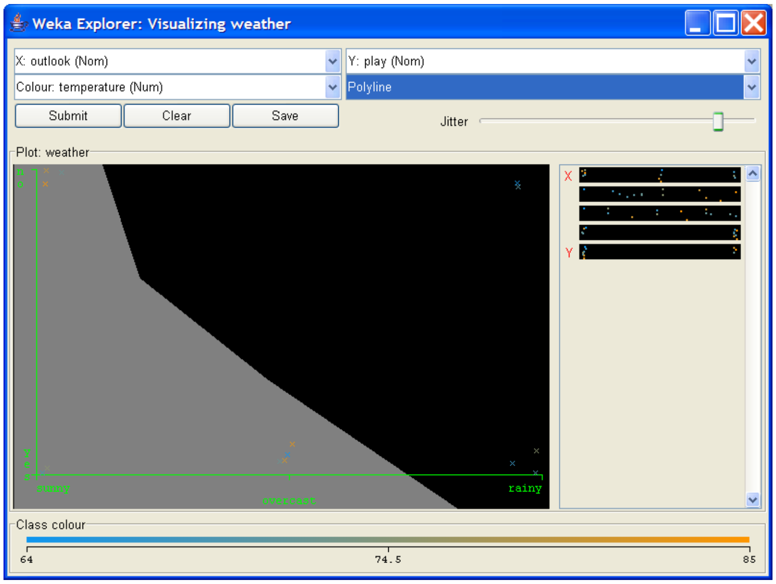
2. **Rectangle**. You can create a rectangle by dragging it around the points.



3. **Polygon**. You can select several points by building a free-form polygon. Left-click on the graph to add vertices to the polygon and right-click to complete it.



4. **Polyline**. To distinguish the points on one side from the once on another, you can build a polyline. Left-click on the graph to add vertices to the polyline and right-click to finish.



Weka Explorer: Visualizing weather

X: outlook (Nom)

Y: play (Nom)

Colour: temperature (Num)

Polyline

Once the area has been selected it is colored gray. You can click on ‘Submit’ button to remove the points outside the gray area. To erase selected (gray) area without affecting the graph, click on ‘Clear’ button. When you clicked on ‘Submit’ button, it changes to ‘Reset’ button. By clicking on ‘Reset’ button, you can undo all changes and restore the original graph. To save all your currently visible instances to ARFF file, click on ‘Save’ button.

9. Conclusion

This concludes WEKA Explorer Tutorial. You have covered a lot of material since the Tutorial Introduction. There is a lot more to learn about WEKA than what you have covered in these seven exercises. But you have already learned enough to be able to analyze your data using preprocessing, classification, clustering, and association rule tools. You have learned how to visualize the result and select attributes. This knowledge will prove invaluable to you. If you plan to do any complicated data analysis, which require software flexibility, I recommend you to use WEKA’s ‘Simple CLI’ interface. So, are you ready yet? Probably not. You have few new tools, but practice makes perfect. Good luck with your data analysis.

10. References

1. Witten, E. Frank, Data Mining, Practical Machine Learning Tools and Techniques with Java Implementation, Morgan Kaufmann Publishers, 2000.
2. R. Kirkby, WEKA Explorer User Guide for version 3-3-4, University of Weikato, 2002.
3. Weka Machine Learning Project, http://www.cs.waikato.ac.nz/~ml/index.html.
4. E.Frank, Machine Learning With WEKA, University of Waikato, New Zealand.
5. B. Mobasher, Data Preparation and Mining with WEKA, http://maya.cs.depaul.edu/~classes/ect584/WEKA/association_rules.html, DePaul University, 2003.
6. M. H. Dunham, Data Mining, Introductory and Advanced Topics, Prentice Hall, 2002.