KARNATAK LAW SOCIETY'S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI – 590008

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(Approved By AICTE, New Delhi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

COURSE PROJECT: DATA MINING

CLASSIFICATION OF GLASS DATASET USING WEKA

Guided by: Prof. Pandurang Upparmani

Hemanth Tubachi	2GI18IS015
Laxmi Nyamagoud	2GI18IS020
Rachana Kampli	2GI18IS032
Rohan Kokatanur	2GI18IS066

2020-2021

CERTIFICATE

This is to certify that **Mr. Hemanth I T, Ms. Laxmi Nyamagoud, Ms. Rachana Kampli, Mr. Rohan Kokatanur** of **Sixth Semester** bearing **USN: 2GI18IS015, 2GI18IS020, 2GI18IS032, 2GI18IS066** has satisfactorily completed the course in Course activity of Data mining. It can be considered as a bonafide work carried out for partial fulfillment of the academic requirement of 6th Semester B.E. (Information Science & Engineering) prescribed by KLS Gogte Institute of Technology, Belagavi during the academic year 2020-21.

The report has been approved as it satisfies the academic requirements prescribed for the said degree.

Signature of The Faculty Member **Signature of the HOD.**

Date: 30/06/2021

Contents

1 **Abstract** 3

2 **Introduction** 3

2.1 **Problem statement** 3

2.2 **Objective** 3

3 **Literature Survey** 3

3.1 **Introduction** 3

3.2 **Launching WEKA Explorer** 4

3.3 **Preprocessing Data** 5

3.3.1 File Conversion 6

3.3.2 Opening file from a local file system 7

3.3.3 Opening file from a web site 8

3.3.4 Reading data from a database 10

3.3.5 Preprocessing window 10

3.3.6 Setting Filters 15

3.4 **Building "Classifiers"** 18

3.4.1 Choosing a Classifier 19

3.4.2 Analyzing Results 23

3.4.3 Visualization of Results 24

4 **Implementation** 27

4.1 **J48 Algorithm** 27

5 **Results** 28

6 **Conclusion** 34

7 **References** 34

1 Abstract

Weka is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. It contains tools for data preparation, classification, regression, clustering, association rules mining, and visualization.

Dataset (or data set) is a collection of data, usually presented in tabular form. Each column represents a particular variable. Each row corresponds to a given member of the dataset in question. It lists values for each of the variables, such as height and weight of an object.

2 Introduction

2.1 Problem statement

The glass dataset glass.arff from the U.S. Forensic Science Service contains data on six types of glass. Glass is described by its refractive index and the chemical elements that it contains; the aim is to classify different types of glass based on these features. This dataset is taken from the UCI datasets, which have been collected by the University of California at Irvine and are freely available on the Web. They are often used as a benchmark for comparing data mining algorithms.

2.2 Objective

- How many attributes are there in the dataset?
- What are their names?
- What is the class attribute?

Run the classification algorithm IBk (weka.classifiers.lazy.IBk). Use cross-validation to test its performance, leaving the number of folds at the default value of 10. Recall that you can examine the classifier options in the Generic Object Editor window that pops up when you click the text beside the Choose button. The default value of the KNN field is 1: This sets the number of neighboring instances.

3 Literature Survey

3.1 Introduction

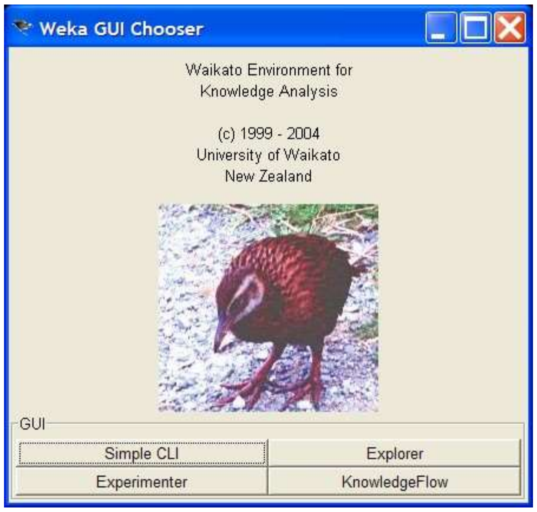
WEKA is a data mining system developed by the University of Waikato in New Zealand that implements data mining algorithms. WEKA is a state-of-the-art facility for developing machine learning (ML) techniques and their application to real-world data mining problems. It is a collection of machine learning 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.

3.2 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' Æ 'Programs' Æ '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]:

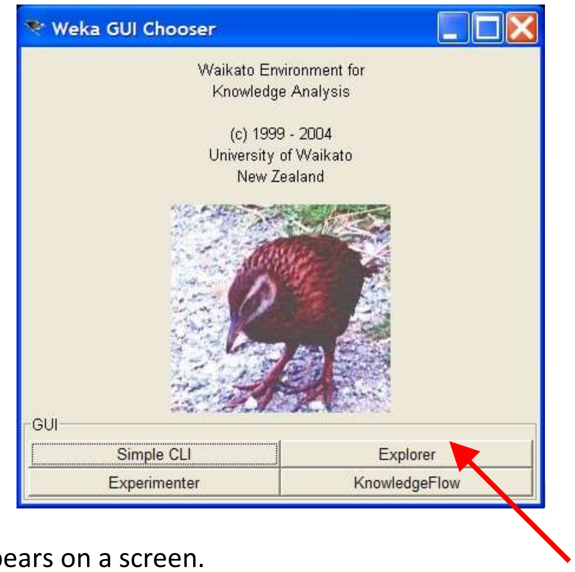


Waikato Environment for Knowledge Analysis

(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.

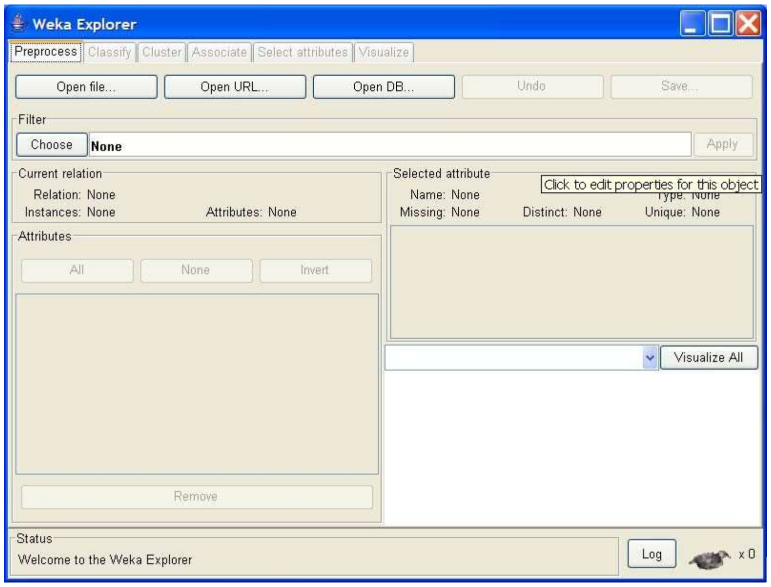
For the exercises in this tutorial you will use 'Explorer'. Click on 'Explorer' button in the 'WEKA GUI Chooser' window.



Waikato Environment for
Knowledge Analysis

(c) 1999 - 2004
University of Waikato
New Zealand

'WEKA Explorer' window appears on a screen.



None

Click to edit properties for this object

Relation: None
Instances: None
Attributes: None

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

Welcome to the Weka Explorer

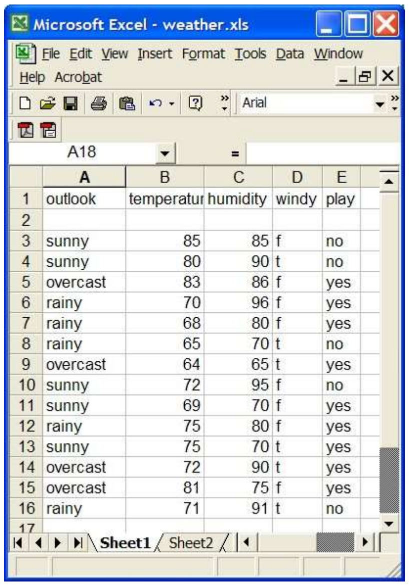
3.3 Preprocessing Data

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.

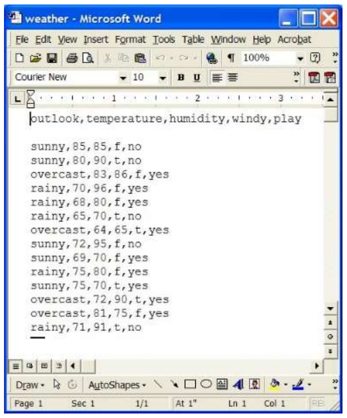
3.3.1 File Conversion

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

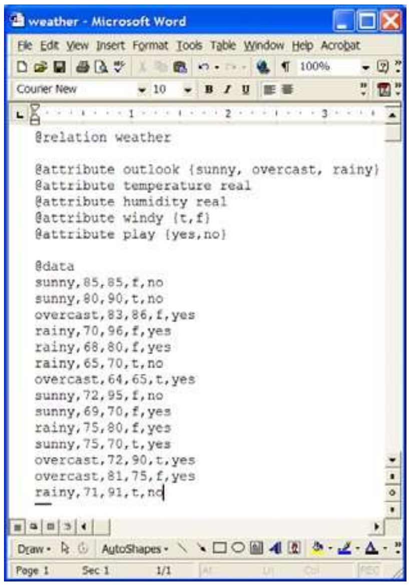
	A	B	C	D	E
1	outlook	temperatur	humidity	windy	play
2					
3	sunny	85	85	f	no
4	sunny	80	90	t	no
5	overcast	83	86	f	yes
6	rainy	70	96	f	yes
7	rainy	68	80	f	yes
8	rainy	65	70	t	no
9	overcast	64	65	t	yes
10	sunny	72	95	f	no
11	sunny	69	70	f	yes
12	rainy	75	80	f	yes
13	sunny	75	70	t	yes
14	overcast	72	90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

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
Courier New
10
Page 1 Sec 1 1/1 At 1" Ln 1 Col 1

```
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.

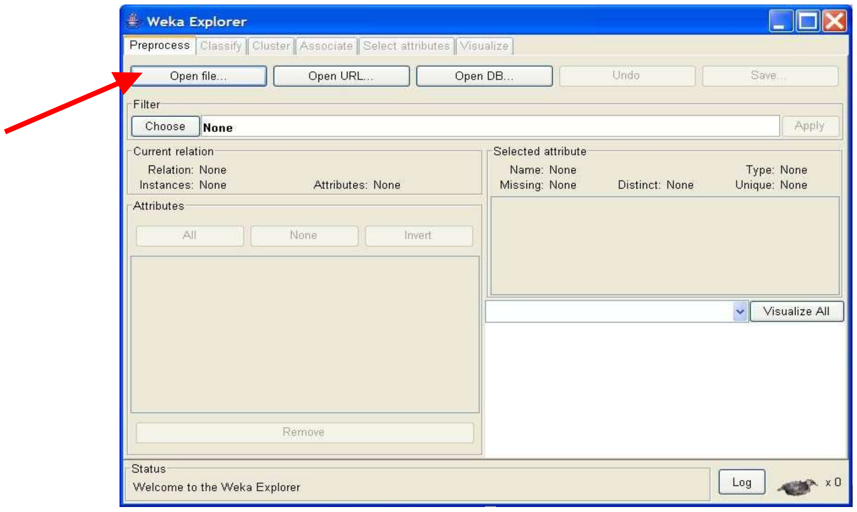


```
@relation weather

@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
```

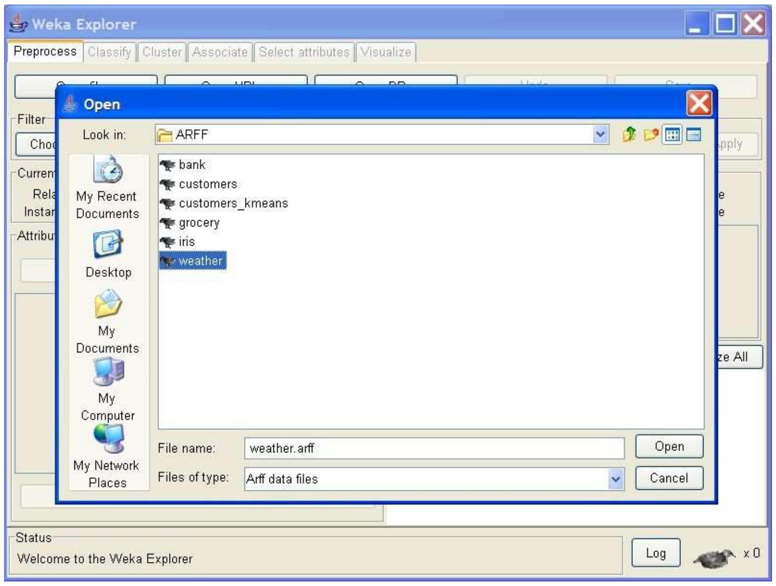
Choose 'Save As…' from the 'File' menu and specify 'Text Only with Line Breaks' as the file type. Enter a file name and click 'Save' button. Rename the file to the file with extension .arff to indicate that it is in ARFF format.

3.3.2 Opening file from a local file system

Click on 'Open file…' button.



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



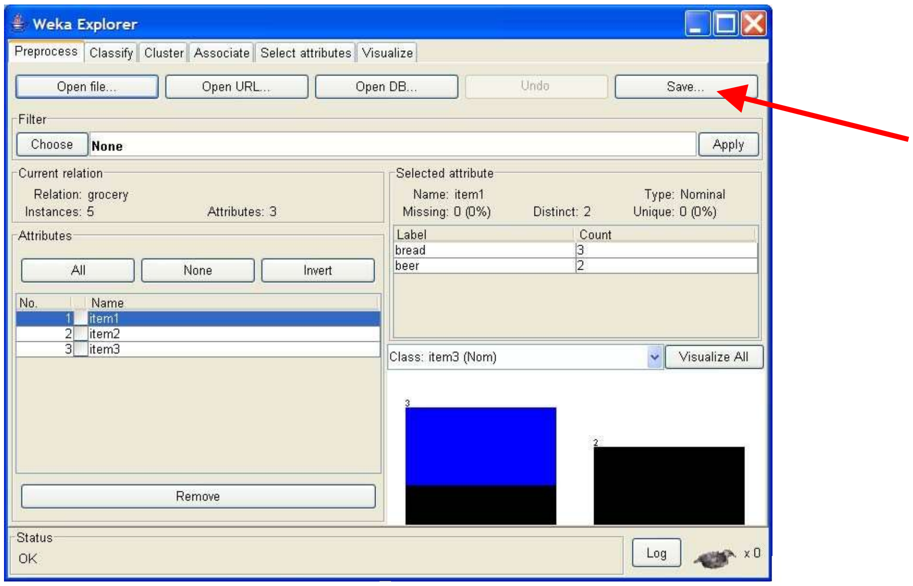
Look in: ARFF

bank
customers
customers_kmeans
grocery
iris
weather

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

No.	Name
1	item1
2	item2
3	item3

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

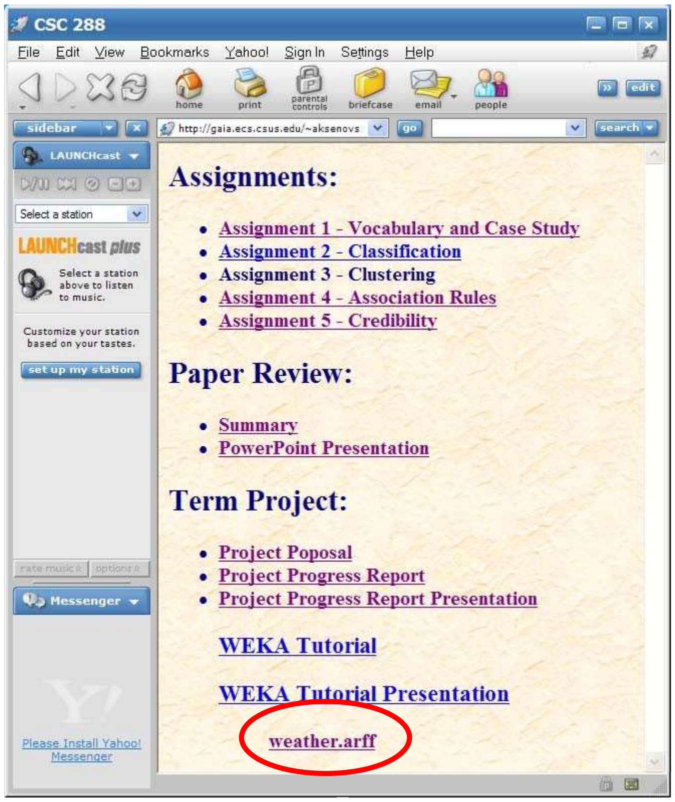
Label	Count
bread	3
beer	2

Class: item3 (Nom)

Status
OK

3.3.3 Opening file from a web site

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


CSC 288
File Edit View Bookmarks Yahoo! Sign In Settings Help
home print parental controls briefcase email people
sidebar
http://gaia.ecs.csus.edu/~aksenovs
go
search
LAUNCHcast
Select a station
LAUNCHcast plus
Select a station above to listen to music.
Customize your station based on your tastes.
set up my station
Messenger
Please Install Yahoo! Messenger

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:

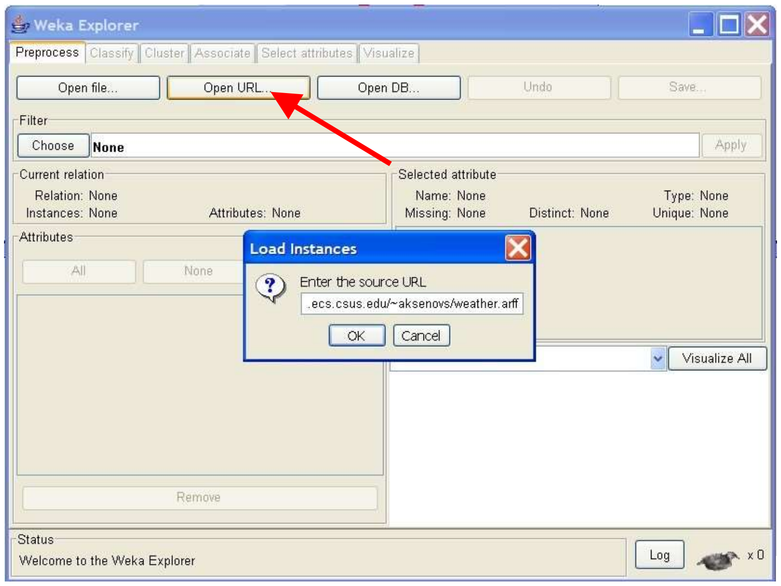
- Project Poposal
- Project Progress Report
- Project Progress Report Presentation

WEKA Tutorial

WEKA Tutorial Presentation

weather.arff

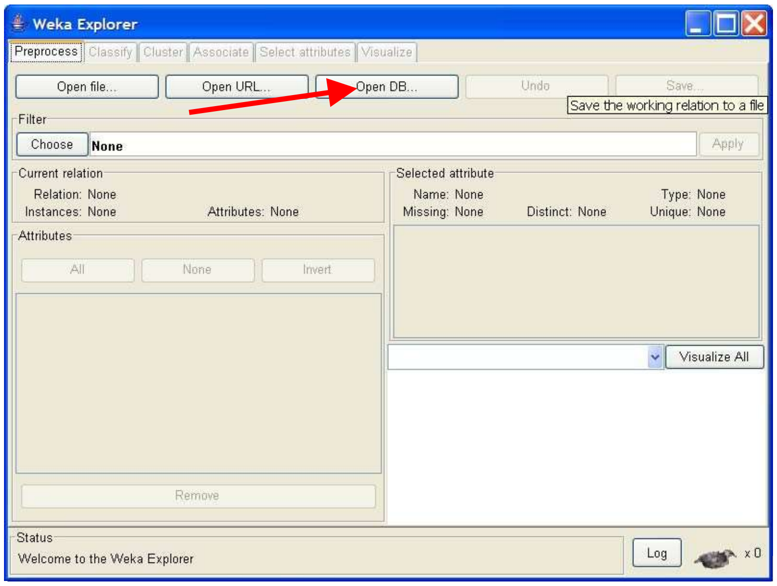
The URL of the web site in our example is http://gaia.ecs.csus.edu/~aksenovs/. It means that the file is stored in this directory, just as in the case with your local file system. To open this file, click on 'Open URL…' button, it brings up a dialog box requesting to enter source URL.


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Undo Save...
Filter
Choose None Apply
Current relation
Relation: None
Instances: None Attributes: None
Selected attribute
Name: None Type: None
Missing: None Distinct: None Unique: None
Attributes
All None
Visualize All
Remove
Status
Welcome to the Weka Explorer
Log x 0
Load Instances
Enter the source URL
.ecs.csus.edu/~aksenovs/weather.arff
OK Cancel

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.

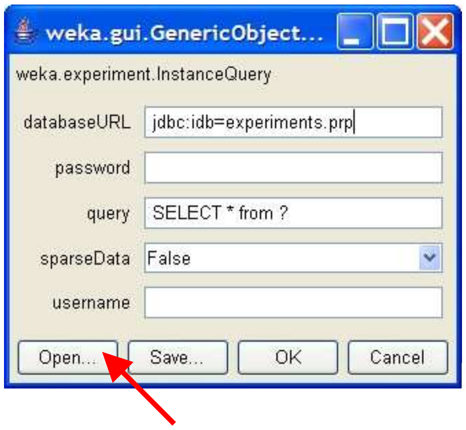
3.3.4 Reading data from a database

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



Relation: None
Instances: None
Attributes: None
Name: None
Missing: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

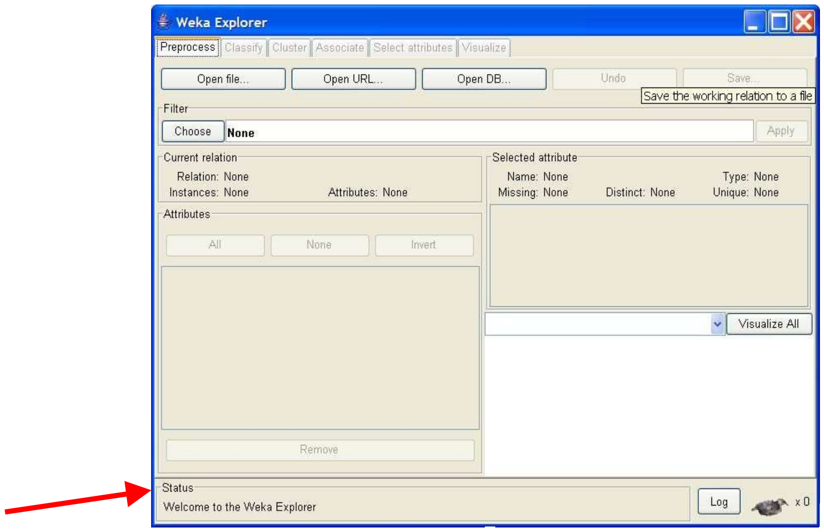


weka.experiment.InstanceQuery

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.

3.3.5 Preprocessing window



Relation: None
Instances: None
Attributes: None
Name: None
Missing: None
Distinct: None
Type: 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

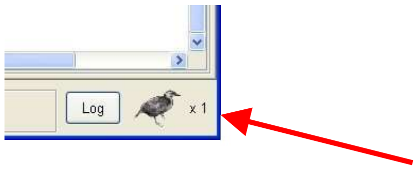
'Explorer', the message says, "Welcome to the Weka Explorer". When you loading

"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.

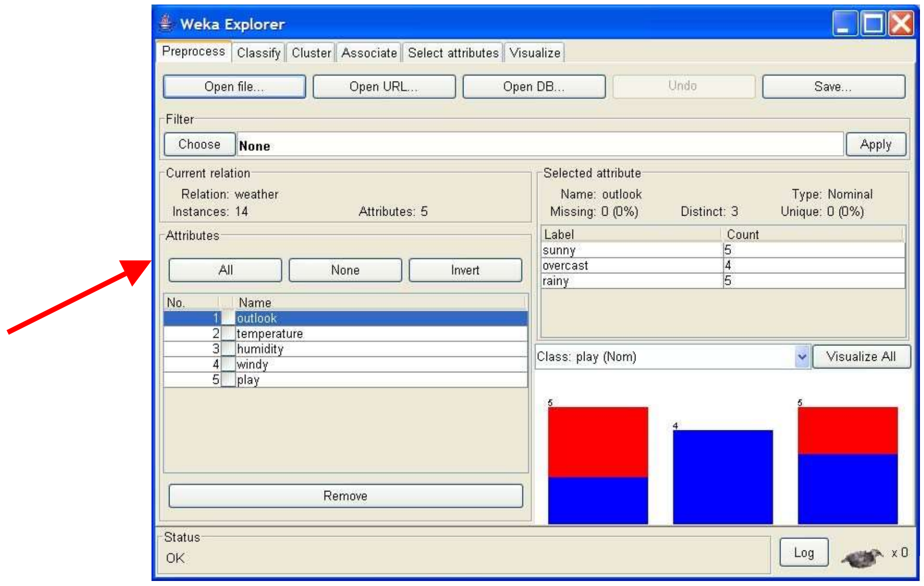
To the right of the 'Log' button there is an image of a bird. The bird is WEKA status icon. The number next to 'X' symbol indicates a number of concurrently running processes. When you loading a file, the bird sits down that means that there are no processes running. The number of processes besides symbol '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.

<u>Loading data</u>
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.



Filter: None

Current relation
Relation: weather
Instances: 14 Attributes: 5

Attributes

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

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

Label	Count
sunny	5
overcast	4
rainy	5

Class: play (Nom)

Status
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,

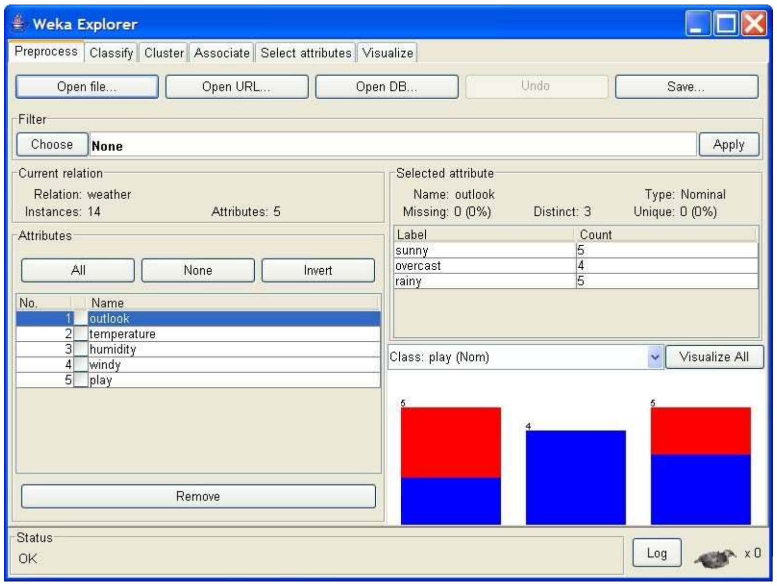
Distinct is the number of different values that the data contains for this attribute, and

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.



Current relation
Relation: weather
Instances: 14 Attributes: 5

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

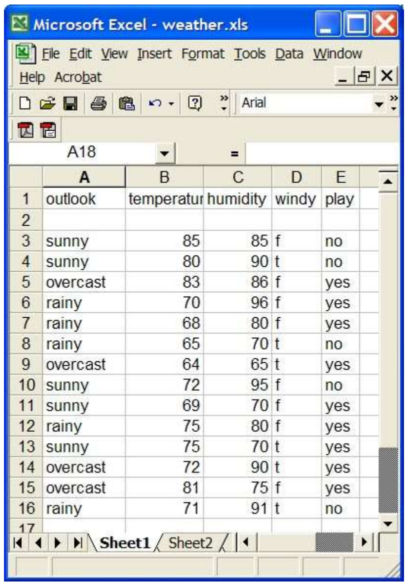
Label	Count
sunny	5
overcast	4
rainy	5

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
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					

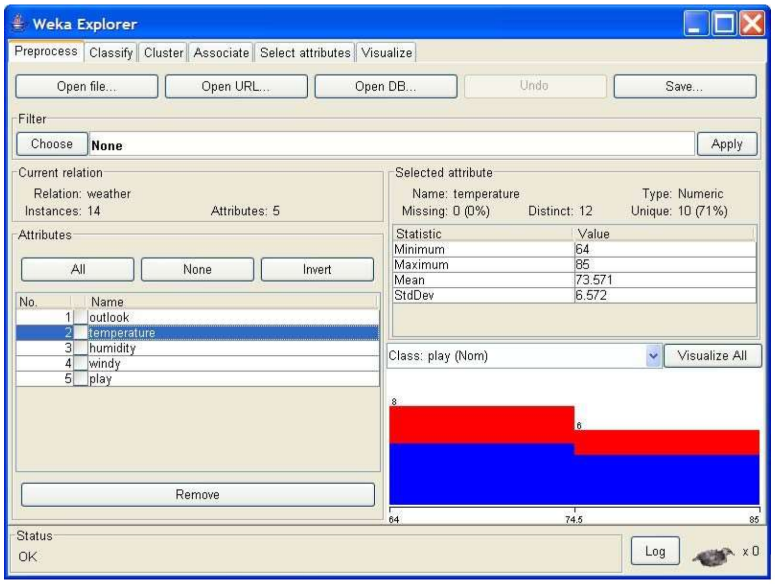
Outlook is nominal. Therefore, you can see the following frequency statistics for this attribute in the 'Selected attributes' window:

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.



Current relation
Relation: weather
Instances: 14 Attributes: 5

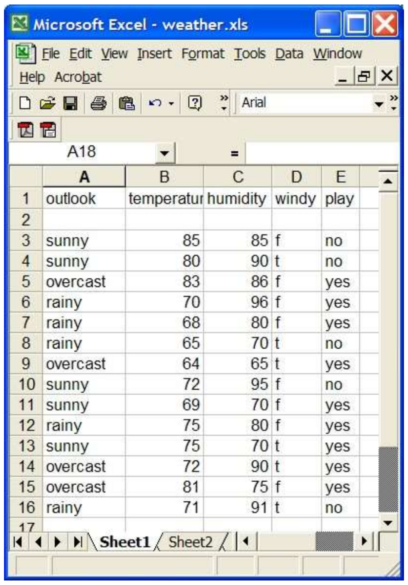
Selected attribute
Name: temperature Type: Numeric
Missing: 0 (0%) Distinct: 12 Unique: 10 (71%)

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)


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					

Temperature is a numeric value; therefore, you can see min, max, means, and standard deviation in ‘Selected Attribute’ window.

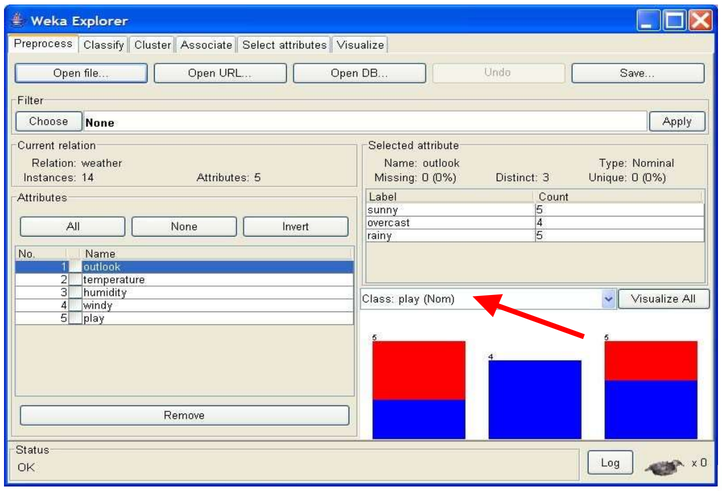
Missing = 0 means that the attribute is specified for all instances (no missing values),

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.



Current relation
Relation: weather
Instances: 14 Attributes: 5

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

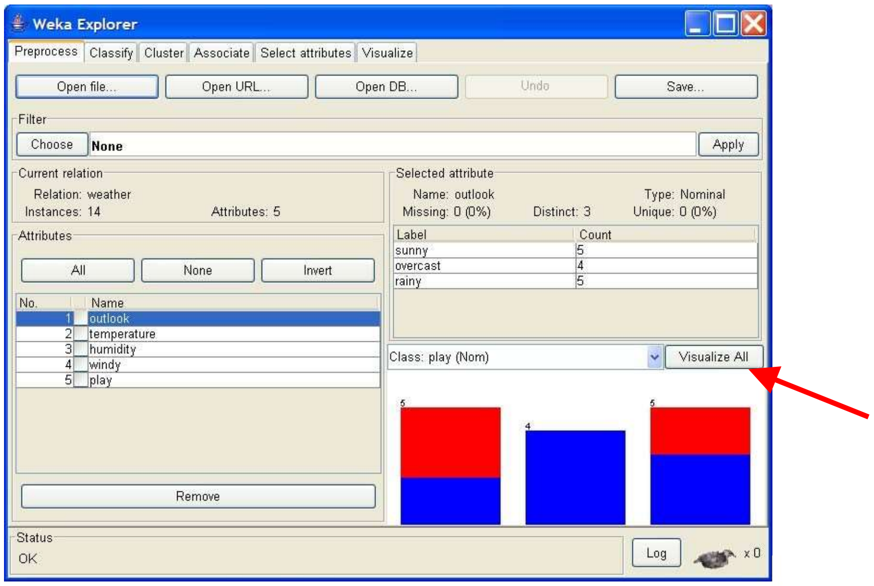
Label	Count
sunny	5
overcast	4
rainy	5

Attributes

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

Class: play (Nom)

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

Current relation
Relation: weather
Instances: 14 Attributes: 5

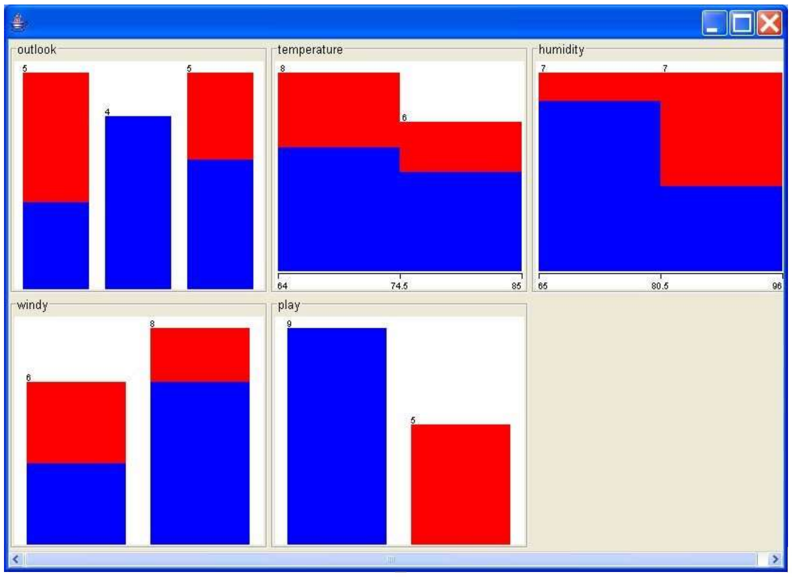
Selected attribute
Name: outlook Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

Label	Count
sunny	5
overcast	4
rainy	5

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

Class: play (Nom)

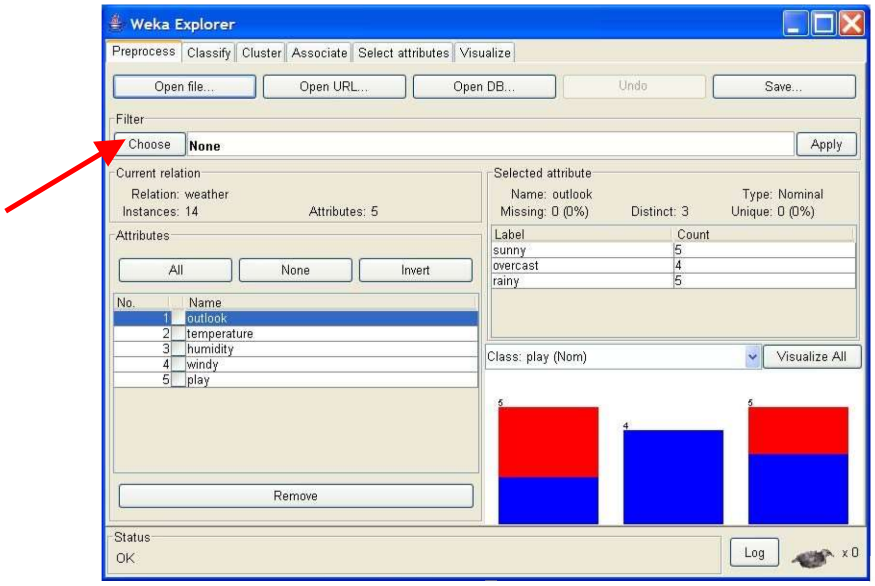
Status
OK



3.3.6 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.



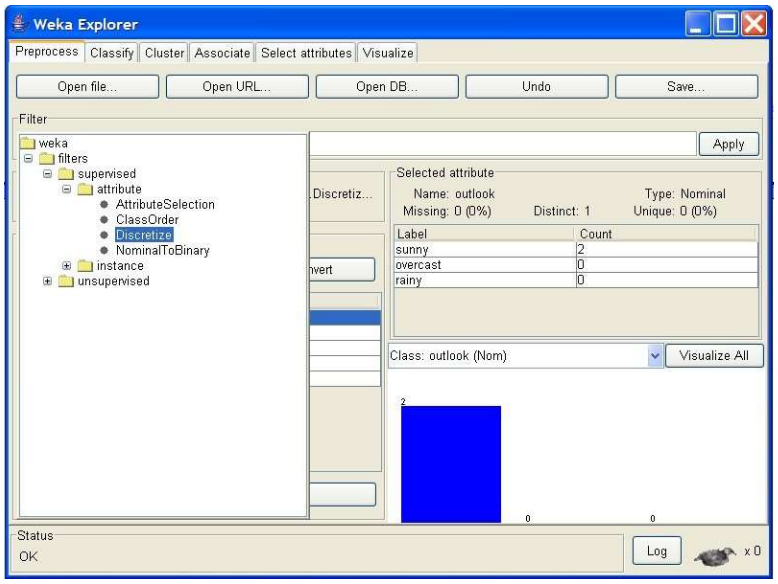
Relation: weather
Instances: 14 Attributes: 5

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

Label	Count
sunny	5
overcast	4
rainy	5

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

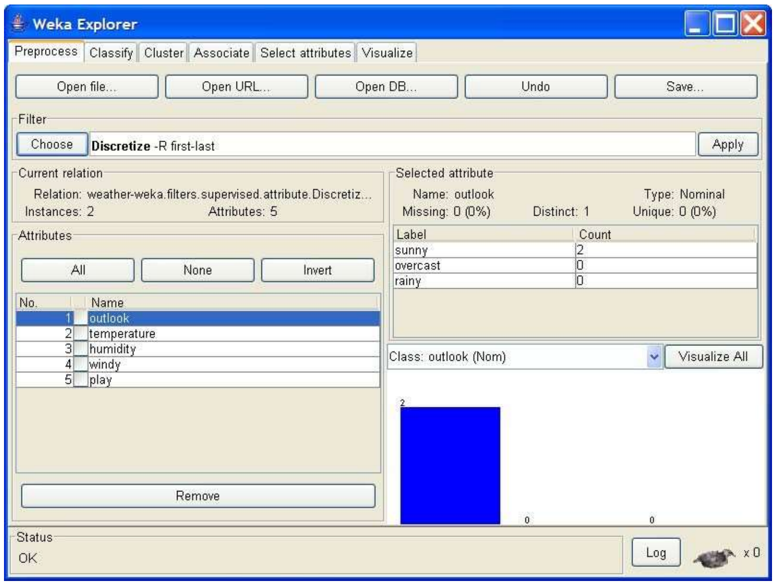
This will show pull-down menu with a list of available filters. Select Supervised Æ Attribute Æ Discretize and click on ‘Apply’ button. The filter will convert Numeric values into Nominal.



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

Label	Count
sunny	2
overcast	0
rainy	0

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

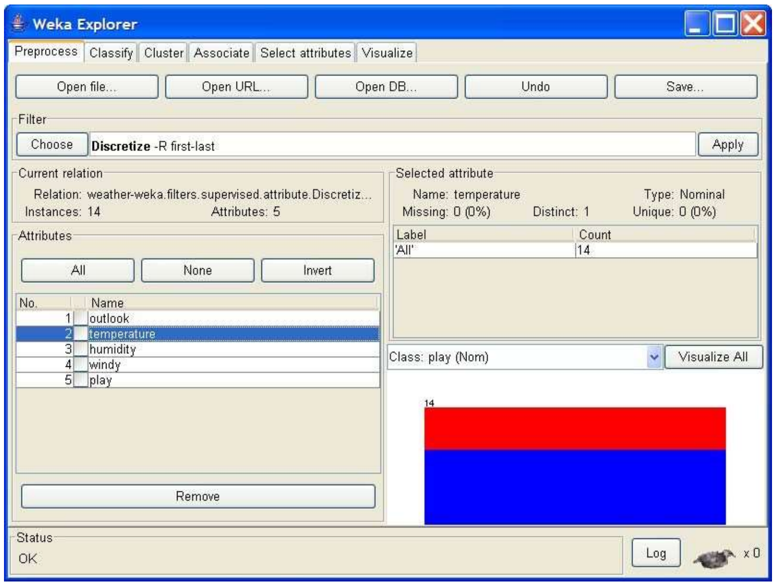
Attributes

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

Attributes

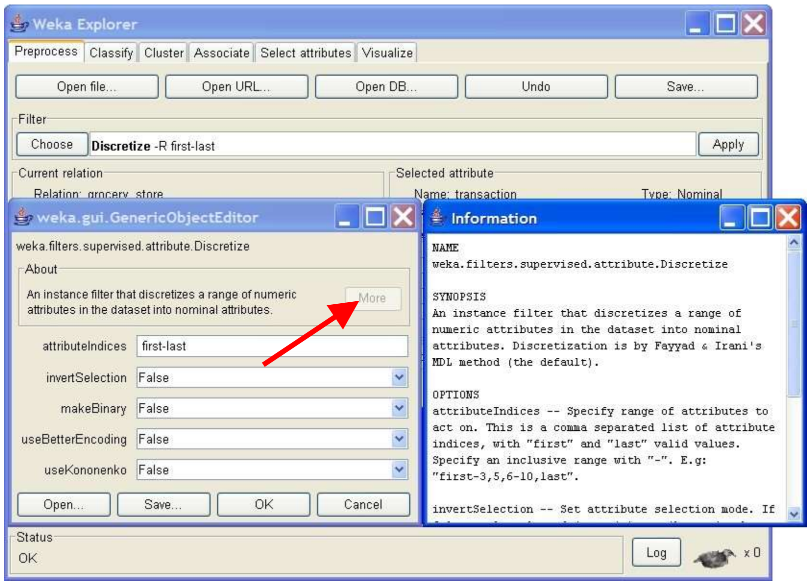
No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

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.



Discretize -R first-last

Relation: grocery_store

Name: transaction   Type: Nominal

weka.filters.supervised.attribute.Discretize

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
```

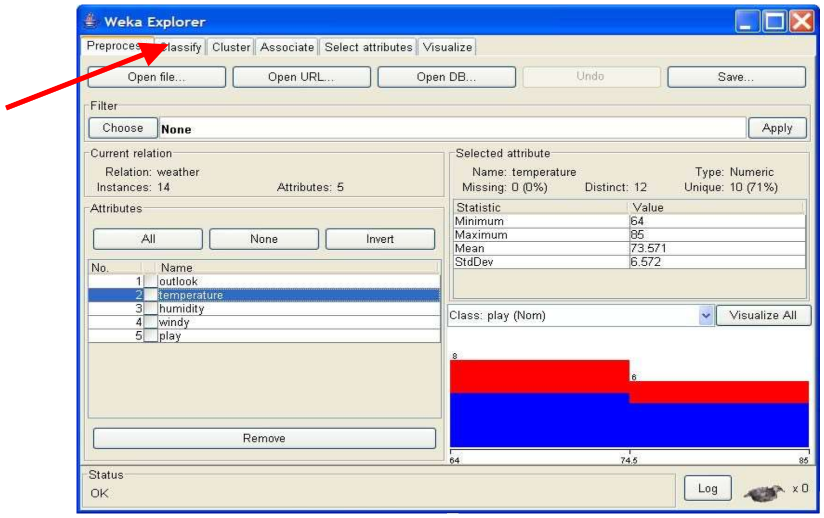
OK

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.

3.4 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.



None

Relation: weather
Instances: 14   Attributes: 5

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

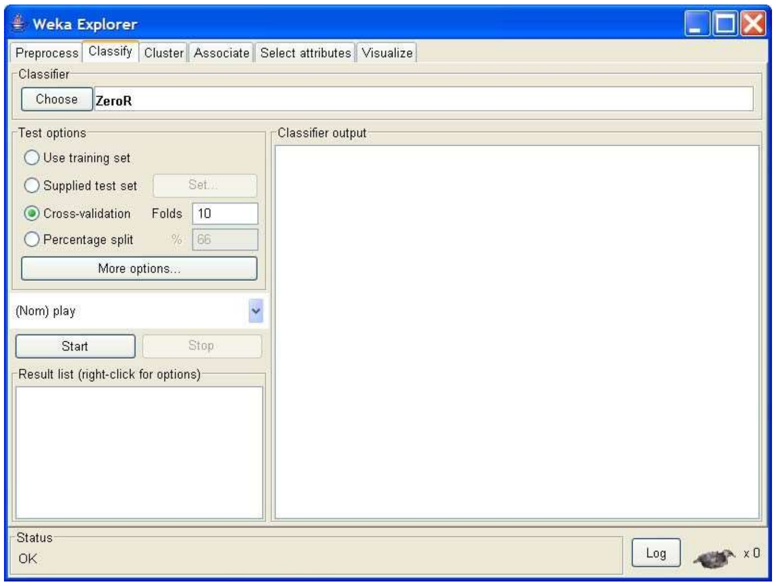
Statistic	Value
Minimum	64
Maximum	85
Mean	73.571
StdDev	6.572

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

Class: play (Nom)

OK

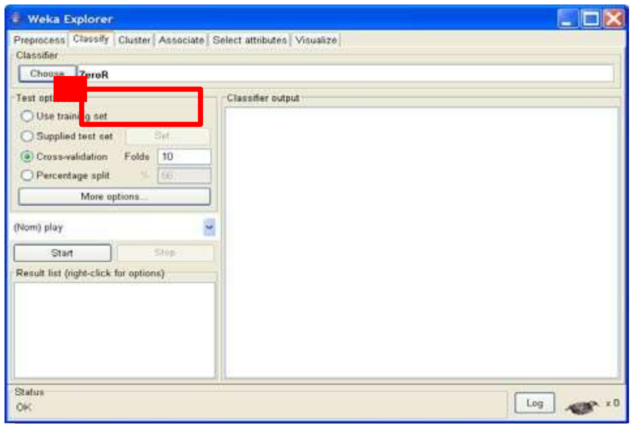
‘Classify’ window comes up on the screen.

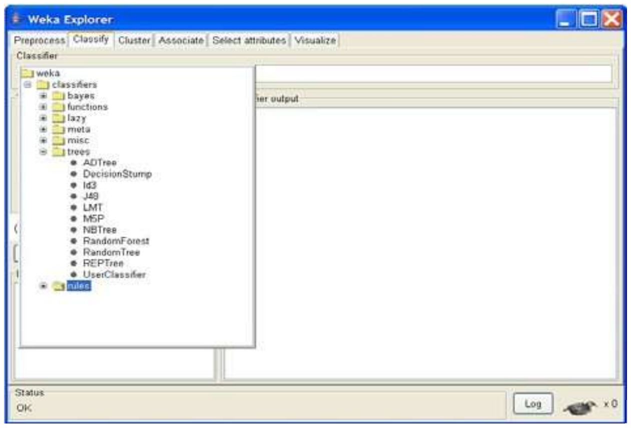


Now you can start analyzing the data using the provided algorithms. In this exercise you will analyze the data with C4.5 algorithm using J48, WEKA’s implementation of decision tree learner. The sample data used in this exercise is the weather data from the file “weather.arff”. Since C4.5 algorithm can 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.

3.4.1 Choosing a Classifier

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



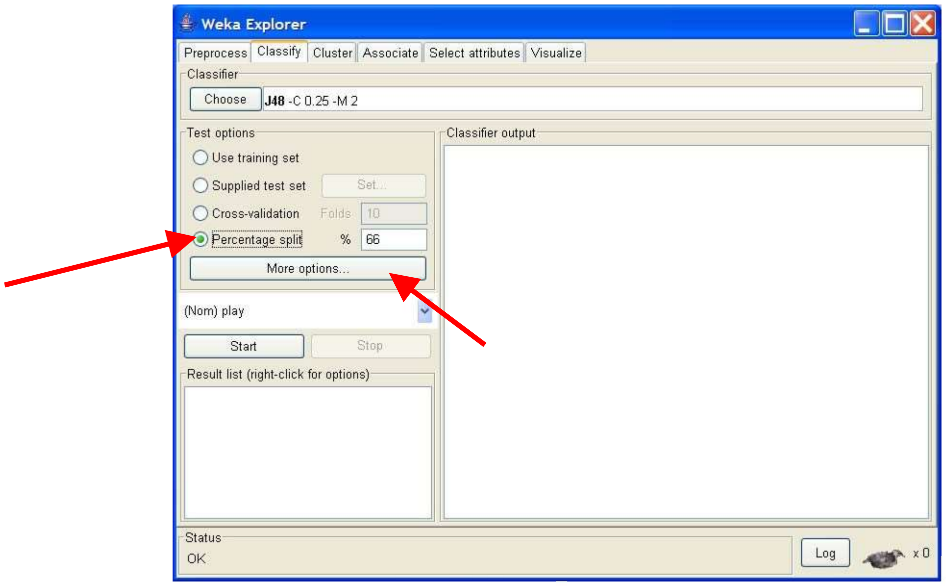


4.2 . Setting Test Options

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

Folds 10

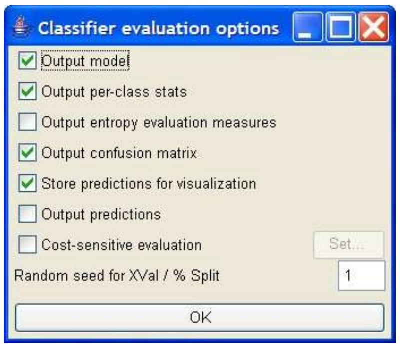
% 66

(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.



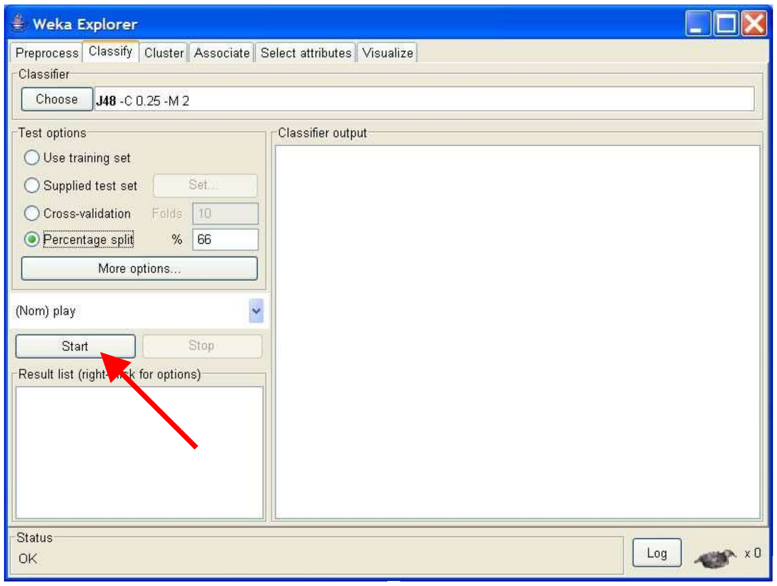
- [✓] Output model
- [✓] Output per-class stats
- [ ] Output entropy evaluation measures
- [✓] Output confusion matrix
- [✓] Store predictions for visualization
- [ ] Output predictions
- [ ] Cost-sensitive evaluation

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.



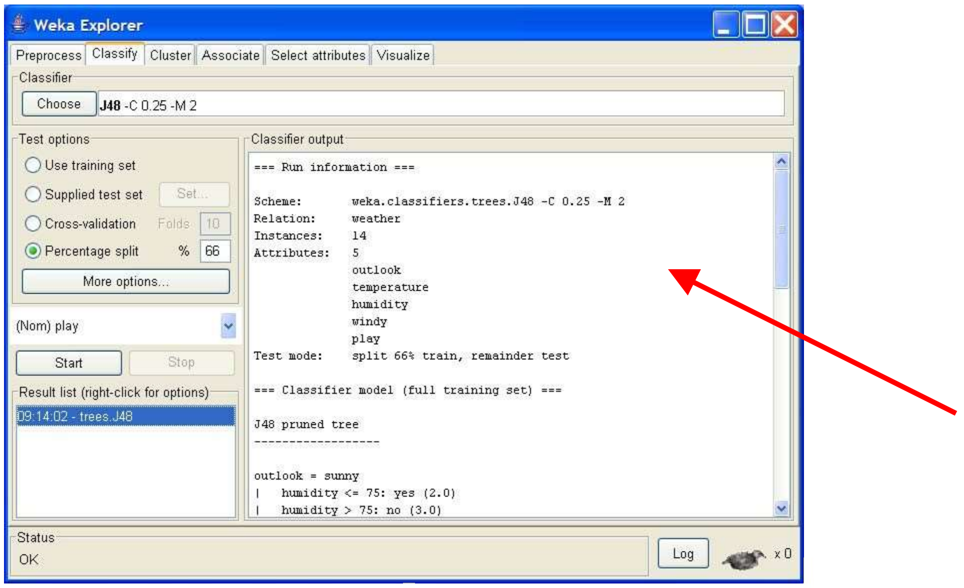
J48 -C 0.25 -M 2

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

(Nom) play

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.



J48 -C 0.25 -M 2

Percentage split % 66

(Nom) play

09:14:02 - trees.J48

```
=== 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)
```

OK

3.4.2 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.

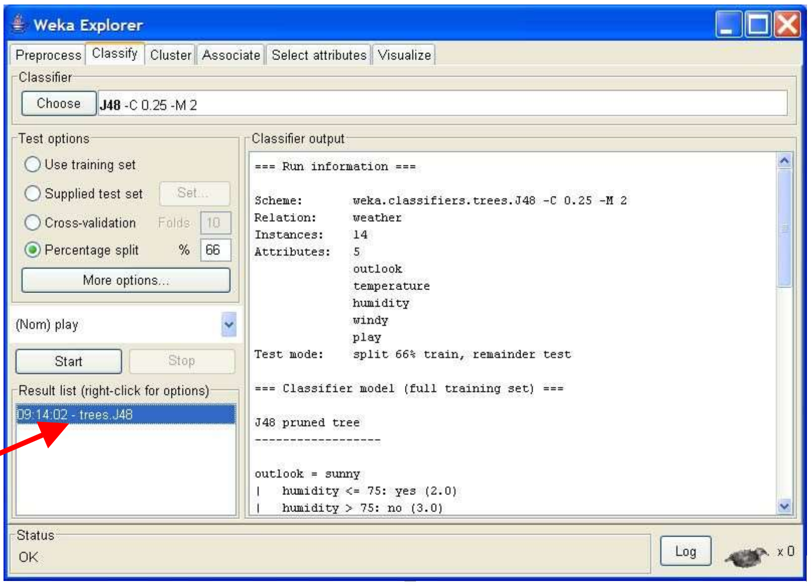
Evaluation on test split. This part of the output gives estimates of the tree's predictive performance, generated by WEKA's evaluation module. It outputs the list of statistics summarizing how accurately the classifier was able to predict the true class of the
instances under the chosen test module. 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 perclass 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'.

3.4.3 Visualization of Results

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



J48 -C 0.25 -M 2

Percentage split % 66

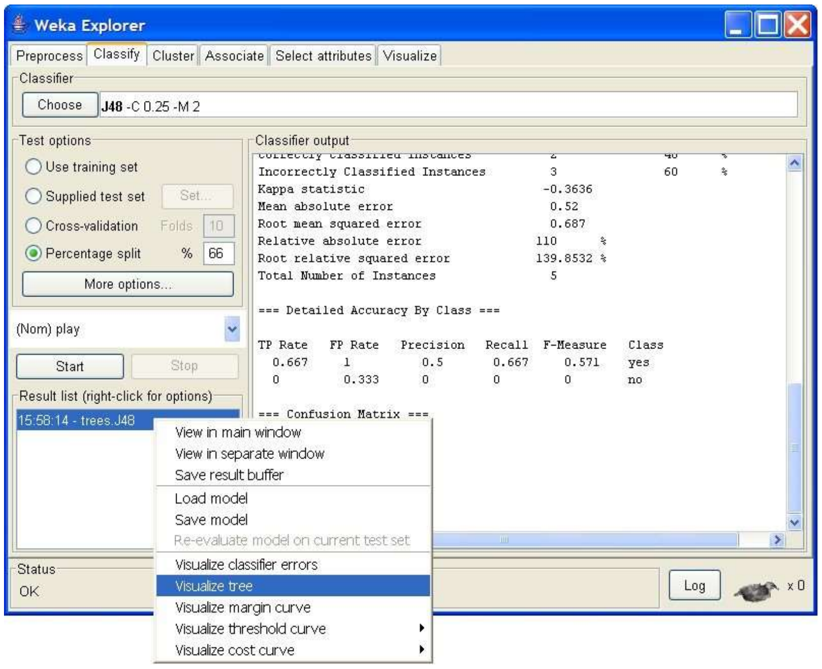
(Nom) play

09:14:02 - trees.J48

```
=== 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)
```

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:



J48 -C 0.25 -M 2

Percentage split % 66

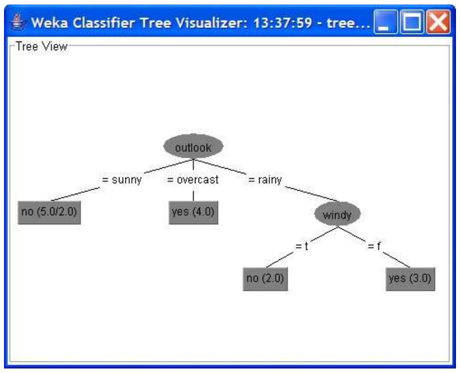
(Nom) play

15:58:14 - trees.J48

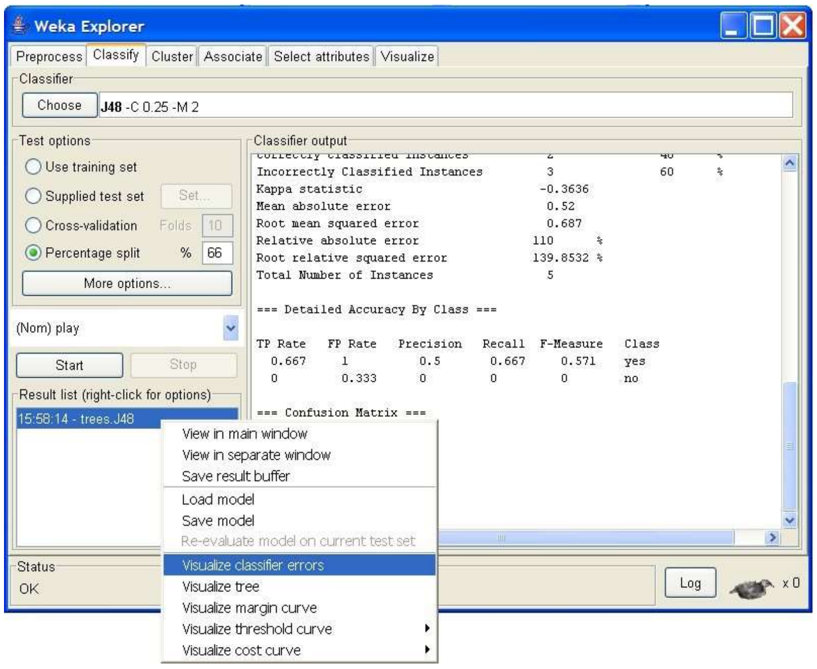
```
Correctly Classified Instances           2               40      %
Incorrectly Classified Instances         3               60      %
Kappa statistic                         -0.3636
Mean absolute error                      0.52
Root mean squared error                  0.687
Relative absolute error                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 ===
```

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:



J48 -C 0.25 -M 2

Folds 10

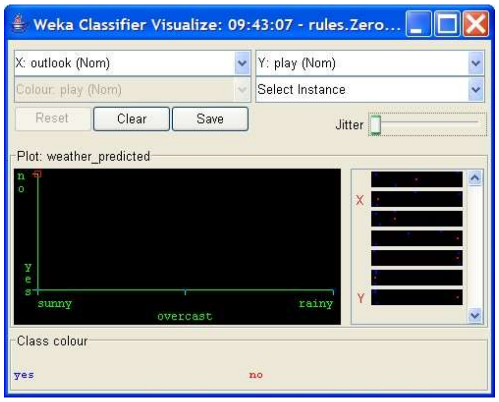
Percentage split % 66

(Nom) play

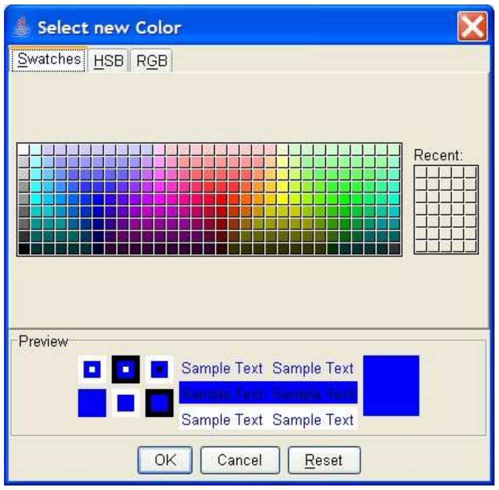
15:58:14 - trees.J48

```
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.

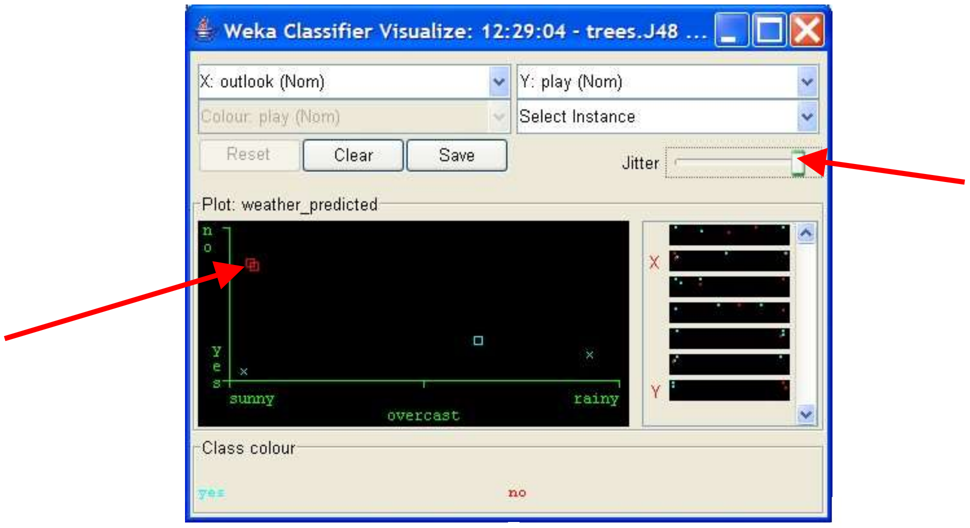


On the ‘Weka Classifier Visualize’ window, beneath the X-axis selector there is a dropdown list, ‘Colour’, for choosing the color scheme. This allows you to choose the color of points based on the attribute selected. Below the plot area, there is a legend that describes what values the colors correspond 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, rightclick 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

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’ Æ 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’ Æ play = ‘no’.

4 Implementation

4.1 J48 Algorithm

C4.5 algorithm actualizes J48 to create a trimmed C4.5 decision tree. Every aspect of the information is to split into minor subsets to base on a decision. J48 look at the standardized data gain that really the results the split the information by choosing an attribute. To summarize, the attribute extreme standardized data gained is utilized. The minor subsets are returned by the algorithm. The split strategies stop if a subset has a place with a similar class in all the instances. J48 develops a decision node utilizing the expected estimations of the class. J48 decision tree can deal with particular characteristics, lost or missing attribute estimations of the data and varying attribute costs. Here accuracy can be expanded by pruning

The Algorithm

Stage 1: The leaf is labeled with a similar class if the instances belong to similar class.

Stage 2: For each attribute, the potential data will be figured and the gain in the data will be taken from the test on the attribute.

Stage 3: Finally the best attribute will be chosen depending upon the current selection parameter

Limitations of J48 Algorithm Despite the fact that J48 one of the well-known algorithms, there are a few shortcomings of this algorithm. A few limitations of J48 are discussed below.

- **Empty Branches** Constructing tree with significant value is one of the important steps for rule generation by J48 algorithm. In our research, we have come out with many nodes with zero values or very close to that. But these values don’t contribute to create or help to create any class for classification task. Instead, it makes the tree wider and still complicating. (Prerna Kapoor, 2015).
- **Insignificant Branches** Number of chosen distinct attributes produces the same number of potential divisions to build a decision tree. But the fact is, not all of them are significant for classification task. These least important branches not only decrease the

usability of decision trees but also bring on the problem of over fitting. (Srishti Taneja, 2014)

- **Over Fitting** Over fitting happens when algorithm display gets information with exceptional attributes. This causes many fragmentations in the process distribution. Statistically unimportant nodes with least examples are known as fragmentations. Usually, J48 algorithm builds trees and grows its branches 'just deep enough to perfectly classify the training examples'. This approach performs better with noise free data. But most of the time this strategy over fits the training examples with noisy data. At present there are two strategies which are widely used to bypass this over fitting in decision tree learning. (SAGAR, 2015) Those are:
  - If tree grows taller, stop it from growing before it reaches the maximum point of accurate classification of the training data.
  - Let the tree to over-fit the training data then post-prune tree.

5 <u>Results</u>

1. **First Run with default parameters.**

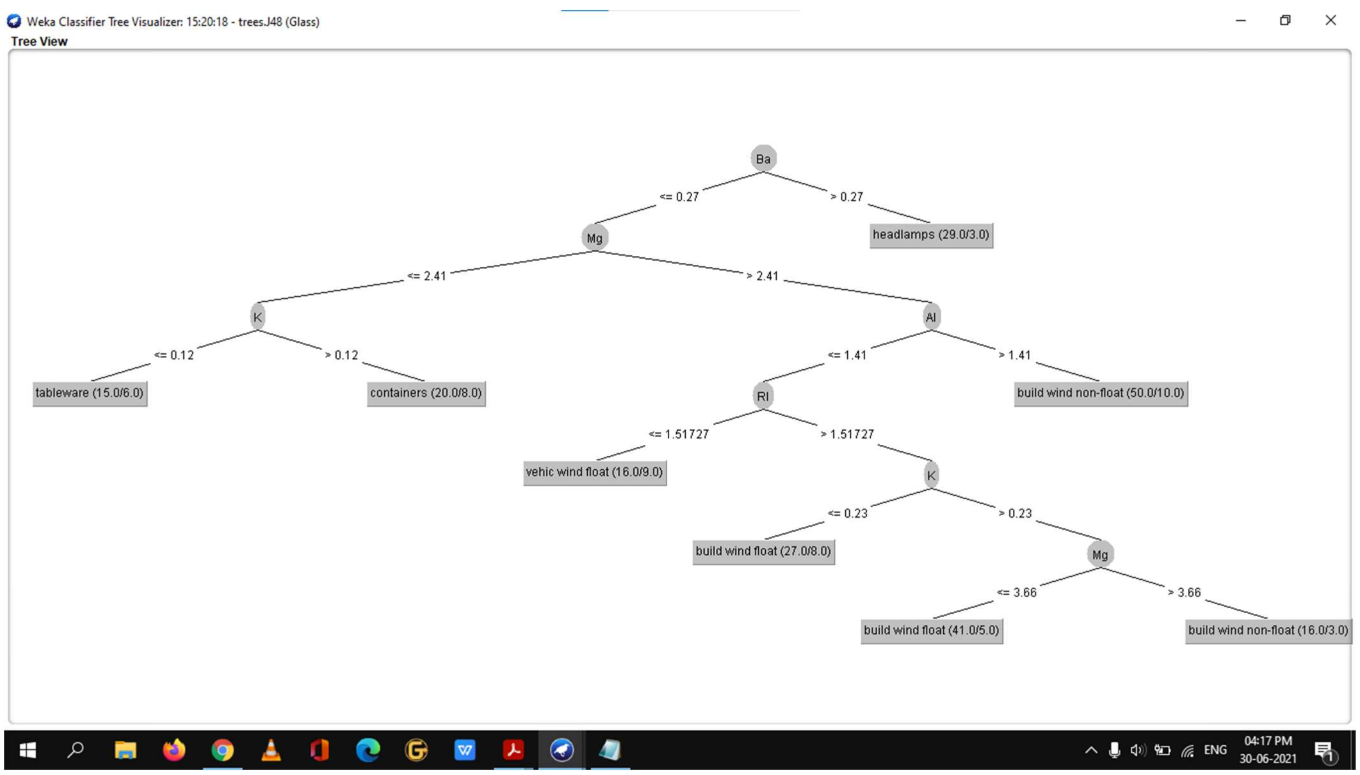
```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     Glass
Instances:    214
Attributes:   10
              RI
              Na
              Mg
              Al
              Si
              K
              Ca
              Ba
              Fe
              Type
Test mode:    10-fold cross-validation

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

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

Ba <= 0.27
|   Mg <= 2.41
|   |   K <= 0.03
|   |   |   Na <= 13.75: build wind non-float (3.0)
|   |   |   Na > 13.75: tableware (9.0)
|   |   K > 0.03
|   |   |   Na <= 13.49
|   |   |   |   RI <= 1.5241: containers (13.0/1.0)
|   |   |   |   RI > 1.5241: build wind non-float (3.0)
|   |   |   Na > 13.49: build wind non-float (7.0/1.0)
|   Mg > 2.41
|   |   Al <= 1.41
|   |   |   RI <= 1.51707
|   |   |   |   RI <= 1.51596: build wind float (3.0)
|   |   |   |   RI > 1.51596
|   |   |   |   |   Fe <= 0.12
|   |   |   |   |   |   Mg <= 3.54: vehic wind float (5.0)
|   |   |   |   |   |   Mg > 3.54
|   |   |   |   |   |   |   RI <= 1.51667: build wind non-float (2.0)
|   |   |   |   |   |   |   RI > 1.51667: vehic wind float (2.0)
|   |   |   |   |   Fe > 0.12: build wind non-float (2.0)
|   |   |   RI > 1.51707
|   |   |   |   K <= 0.23
|   |   |   |   |   Mg <= 3.34: build wind non-float (2.0)
|   |   |   |   |   Mg > 3.34
|   |   |   |   |   |   Si <= 72.64
|   |   |   |   |   |   |   Na <= 14.01: build wind float (14.0)
|   |   |   |   |   |   |   Na > 14.01
|   |   |   |   |   |   |   |   RI <= 1.52211
|   |   |   |   |   |   |   |   |   Na <= 14.32: vehic wind float (3.0)
|   |   |   |   |   |   |   |   |   Na > 14.32: build wind float (2.0)
|   |   |   |   |   |   |   |   RI > 1.52211: build wind float (3.0)
|   |   |   |   |   |   Si > 72.64: vehic wind float (3.0)
|   |   |   |   K > 0.23
|   |   |   |   |   Mg <= 3.75
|   |   |   |   |   |   Fe <= 0.14
|   |   |   |   |   |   |   RI <= 1.52043: build wind float (36.0)
|   |   |   |   |   |   |   RI > 1.52043: build wind non-float (2.0/1.0)
|   |   |   |   |   |   Fe > 0.14
|   |   |   |   |   |   |   Al <= 1.17: build wind non-float (5.0)
|   |   |   |   |   |   |   Al > 1.17: build wind float (6.0/1.0)
|   |   |   |   |   Mg > 3.75: build wind non-float (10.0)
|   |   Al > 1.41
|   |   |   Si <= 72.49
|   |   |   |   Ca <= 8.28: build wind non-float (6.0)
|   |   |   |   Ca > 8.28: vehic wind float (5.0/1.0)
|   |   |   Si > 72.49
|   |   |   |   RI <= 1.51732
|   |   |   |   |   Fe <= 0.22: build wind non-float (30.0/1.0)
|   |   |   |   |   Fe > 0.22
|   |   |   |   |   |   RI <= 1.51629: build wind float (2.0)
|   |   |   |   |   |   RI > 1.51629: build wind non-float (2.0)
|   |   |   |   RI > 1.51732
|   |   |   |   |   RI <= 1.51789: build wind float (3.0)
|   |   |   |   |   RI > 1.51789: build wind non-float (2.0)
Ba > 0.27
|   Si <= 70.16: build wind non-float (2.0/1.0)
|   Si > 70.16: headlamps (27.0/1.0)

Number of Leaves  :       30

Size of the tree :        59


Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        143             66.8224 %
Incorrectly Classified Instances       71             33.1776 %
Kappa statistic                         0.55
Mean absolute error                     0.1026
Root mean squared error                 0.2897
Relative absolute error                48.4507 %
Root relative squared error            89.2727 %
Total Number of Instances             214

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.714    0.174    0.667      0.714    0.690      0.532    0.806     0.667     build wind float
               0.618    0.181    0.653      0.618    0.635      0.443    0.768     0.606     build wind non-float
               0.353    0.046    0.400      0.353    0.375      0.325    0.766     0.251     vehic wind float
               ?        0.000    ?          ?        ?          ?        ?         ?         vehic wind non-float
               0.769    0.010    0.833      0.769    0.800      0.788    0.872     0.575     containers
               0.778    0.029    0.538      0.778    0.636      0.629    0.930     0.527     tableware
               0.793    0.022    0.852      0.793    0.821      0.795    0.869     0.738     headlamps
Weighted Avg.  0.668    0.130    0.670      0.668    0.668      0.539    0.807     0.611

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 50 15  3  0  0  1  1 |  a = build wind float
 16 47  6  0  2  3  2 |  b = build wind non-float
  5  5  6  0  0  1  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  2  0  0 10  0  1 |  e = containers
  1  1  0  0  0  7  0 |  f = tableware
  3  2  0  0  0  1 23 |  g = headlamps
```

2. Parameters Tuned Run.

```
=== Run information ===

Scheme:      weka.classifiers.trees.J48 -U -M 15
Relation:    Glass
Instances:   214
Attributes:  10
             RI
             Na
             Mg
             Al
             Si
             K
             Ca
             Ba
             Fe
             Type
Test mode:   10-fold cross-validation

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

J48 unpruned tree
------------------

Ba <= 0.27
|   Mg <= 2.41
|   |   K <= 0.12: tableware (15.0/6.0)
|   |   K > 0.12: containers (20.0/8.0)
|   Mg > 2.41
|   |   Al <= 1.41
|   |   |   RI <= 1.51727: vehic wind float (16.0/9.0)
|   |   |   RI > 1.51727
|   |   |   |   K <= 0.23: build wind float (27.0/8.0)
|   |   |   |   K > 0.23
|   |   |   |   |   Mg <= 3.66: build wind float (41.0/5.0)
|   |   |   |   |   Mg > 3.66: build wind non-float (16.0/3.0)
|   |   Al > 1.41: build wind non-float (50.0/10.0)
Ba > 0.27: headlamps (29.0/3.0)

Number of Leaves  :      8

Size of the tree :    15


Time taken to build model: 0 seconds
```

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        131             61.215  %
Incorrectly Classified Instances       83             38.785  %
Kappa statistic                      0.4757
Mean absolute error                    0.1292
Root mean squared error                  0.2728
Relative absolute error              61.0227 %
Root relative squared error            84.0521 %
Total Number of Instances             214

=== Detailed Accuracy By Class ===

             TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
             0.614    0.208    0.589      0.614    0.601      0.402    0.831     0.649     build wind float
             0.539    0.225    0.569      0.539    0.554      0.319    0.733     0.636     build wind non-float
             0.118    0.020    0.333      0.118    0.174      0.159    0.745     0.223     vehic wind float
             ?        0.000    ?          ?        ?          ?        ?         ?         vehic wind non-float
             0.846    0.030    0.647      0.846    0.733      0.721    0.934     0.529     containers
             0.889    0.039    0.500      0.889    0.640      0.649    0.921     0.401     tableware
             0.897    0.022    0.867      0.897    0.881      0.863    0.912     0.726     headlamps
Weighted Avg.    0.612    0.156    0.599      0.612    0.598      0.445    0.811     0.603

=== Confusion Matrix ===

  a  b  c  d  e  f  g   <-- classified as
 43 25  1  0  0  0  1 |  a = build wind float
 19 41  3  0  5  6  2 |  b = build wind non-float
  9  6  2  0  0  0  0 |  c = vehic wind float
  0  0  0  0  0  0  0 |  d = vehic wind non-float
  0  0  0  0 11  1  1 |  e = containers
  1  0  0  0  0  8  0 |  f = tableware
  1  0  0  0  1  1 26 |  g = headlamps
```

3. Decision Tree Formed


Weka Classifier Tree Visualizer: 15:20:18 - trees.J48 (Glass)
Tree View
Ba
<= 0.27
> 0.27
Mg
headlamps (29.0/3.0)
<= 2.41
> 2.41
K
Al
<= 0.12
> 0.12
<= 1.41
> 1.41
tableware (15.0/6.0)
containers (20.0/8.0)
RI
build wind non-float (50.0/10.0)
<= 1.51727
> 1.51727
vehic wind float (16.0/9.0)
K
<= 0.23
> 0.23
build wind float (27.0/8.0)
Mg
<= 3.66
> 3.66
build wind float (41.0/5.0)
build wind non-float (16.0/3.0)
ENG
04:17 PM
30-06-2021

6 Conclusion

- In general, to mining knowledge from the data, we first need to determine the objective, what do we want. From the data, we can make overview, describe the data. After that, we can use other advanced technique to make analysis.
- Analysis and visualize the result are very efficient for decision maker. it is fast and accurate to know the information.
- The disadvantage of those methods is that, it is difficult for the person who directly work with the data, the person who use these techniques. He/she must be expert with algorithms of each method, really understanding the data to fit with each technique.
- In addition, when the data is huge, it is very difficult to visualize the data, the graph, diagram, plot can be over crowded.

7 References

1. Dataset Used: glass.arff
2. WEKA Explorer Tutorial.doc School of Engineering and Computer Science Department of Computer Science California State University, Sacramento California, 95819.
3. https://cobweb.cs.uga.edu/~khaled/DMcourse/Weka-Tutorial-Exercises.pdf.
4. https://www.periyaruniversity.ac.in/ijcii/issue/marnew/2_mar_18.pdf.
5. As usual, Google and Wikipedia.