LABORATORY MANUAL
DATA WAREHOUSING AND MINING LAB

B.TECH
(III YEAR – II SEM)
(2018-19)

DEPARTMENT OF INFORMATION TECHNOLOGY

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(Autonomous Institution – UGC, Govt. of India)

Recognized under 2(f) and 12 (B) of UGC ACT 1956
Affiliated to JNTUH, Hyderabad, Approved by AICTE - Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2008 Certified)
Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad – 500100, Telangana State, India

DEPARTMENT OF INFORMATION TECHNOLOGY

Vision

- To acknowledge quality education and instill high patterns of discipline making the students technologically superior and ethically strong which involves the improvement in the quality of life in human race.

Mission

- To achieve and impart holistic technical education using the best of infrastructure, outstanding technical and teaching expertise to establish the students into competent and confident engineers.
- Evolving the center of excellence through creative and innovative teaching learning practices for promoting academic achievement to produce internationally accepted competitive and world class professionals.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1 – ANALYTICAL SKILLS

1. To facilitate the graduates with the ability to visualize, gather information, articulate, analyze, solve complex problems, and make decisions. These are essential to address the challenges of complex and computation intensive problems increasing their productivity.

PEO2 – TECHNICAL SKILLS

2. To facilitate the graduates with the technical skills that prepare them for immediate employment and pursue certification providing a deeper understanding of the technology in advanced areas of computer science and related fields, thus encouraging to pursue higher education and research based on their interest.

PEO3 – SOFT SKILLS

3. To facilitate the graduates with the soft skills that include fulfilling the mission, setting goals, showing self-confidence by communicating effectively, having a positive attitude, get involved in team-work, being a leader, managing their career and their life.

PEO4 – PROFESSIONAL ETHICS

To facilitate the graduates with the knowledge of professional and ethical responsibilities by paying attention to grooming, being conservative with style, following dress codes, safety codes, and adapting themselves to technological advancements.

PROGRAM SPECIFIC OUTCOMES (PSOs)

After the completion of the course, B. Tech Information Technology, the graduates will have the following Program Specific Outcomes:

1. **Fundamentals and critical knowledge of the Computer System:-** Able to Understand the working principles of the computer System and its components , Apply the knowledge to build, asses, and analyze the software and hardware aspects of it .

2. **The comprehensive and Applicative knowledge of Software Development:** Comprehensive skills of Programming Languages, Software process models, methodologies, and able to plan, develop, test, analyze, and manage the software and hardware intensive systems in heterogeneous platforms individually or working in teams.

3. **Applications of Computing Domain & Research:** Able to use the professional, managerial, interdisciplinary skill set, and domain specific tools in development processes, identify the research gaps, and provide innovative solutions to them.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design / development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication** : Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance** : Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi disciplinary environments.
12. **Life- long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

Maisammaguda, Dhulapally Post, Via Hakimpet, Secunderabad – 500100

DEPARTMENT OF INFORMATION TECHNOLOGY

GENERAL LABORATORY INSTRUCTIONS

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with:

a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session.
b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab.
c. Proper Dress code and Identity card.

4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty.
5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintain the discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions.Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out ; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leaves the lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Head of the Department **Principal**

COURSE NAME: DATA WAREHOUSING AND MINING LAB

COURSE CODE: R15A0590

COURSE OBJECTIVES:

1. Learn how to build a data warehouse and query it (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics).
2. Learn to perform data mining tasks using a data mining toolkit (such as open source WEKA).
3. Understand the data sets and data preprocessing.
4. Demonstrate the working of algorithms for data mining tasks such association rule mining, classification, clustering and regression.
5. Exercise the data mining techniques with varied input values for different parameters.
6. To obtain Practical Experience Working with all real data sets.
7. Emphasize hands-on experience working with all real data sets.

COURSE OUTCOMES:

1. Ability to understand the various kinds of tools.
2. Demonstrate the classification, clustering and etc. in large data sets.
3. Ability to add mining algorithms as a component to the exiting tools.
4. Ability to apply mining techniques for realistic data.

MAPPING OF COURSE OUTCOMES WITH PROGRAM OUTCOMES:

COURSE OUTCOMES	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	P09	PO10	PO11
• Ability to add mining algorithms as a component to the exiting tools. • Ability to apply mining techniques for realistic data.	√	√	√								√

DATA WAREHOUSING AND MINING LAB- INDEX

WEEK -1

Explore visualization features of the tool for analysis like identifying trends etc.

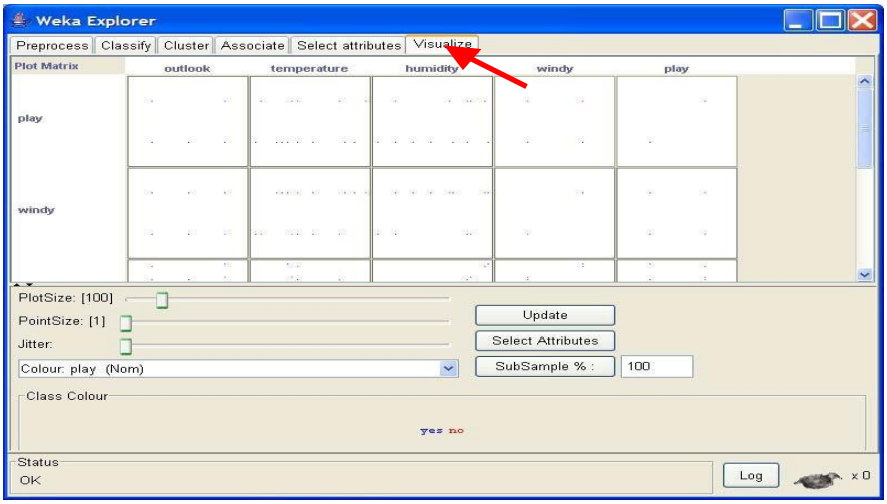
Ans:

Visualization Features:

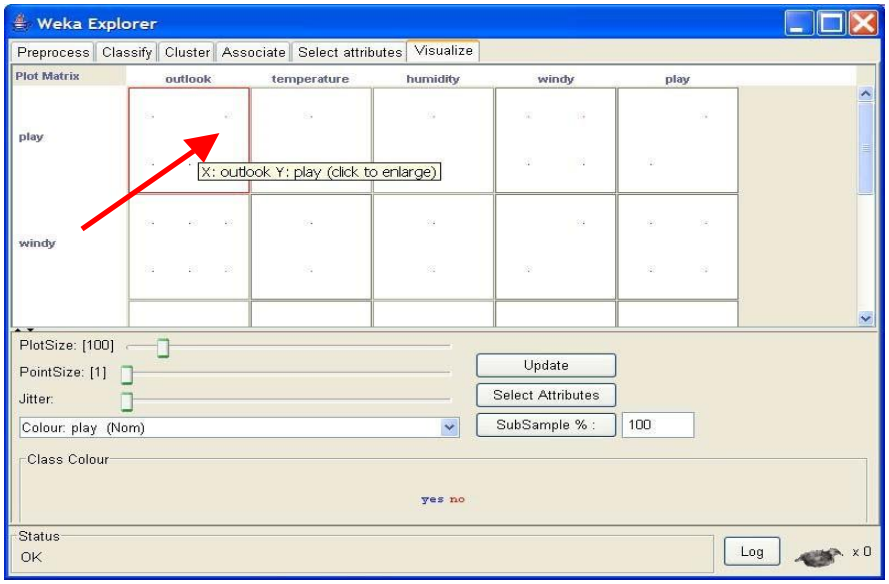
WEKA's visualization allows you to visualize a 2-D plot of the current working relation. Visualization is very useful in practice, it helps to determine difficulty of the learning problem. WEKA can visualize single attributes (1-d) and pairs of attributes (2-d), rotate 3-d visualizations (Xgobi-style).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

- Access To **Visualization** From The *Classifier*, *Cluster* And *Attribute Selection* Panel Is Available From A Popup Menu. Click The Right Mouse Button Over An Entry In The Result List To Bring Up The Menu. You Will Be Presented With Options For Viewing Or Saving The Text Output And --- Depending On The Scheme --- Further Options For Visualizing Errors, Clusters, Trees Etc.

To open Visualization screen, click 'Visualize' tab.



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



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.

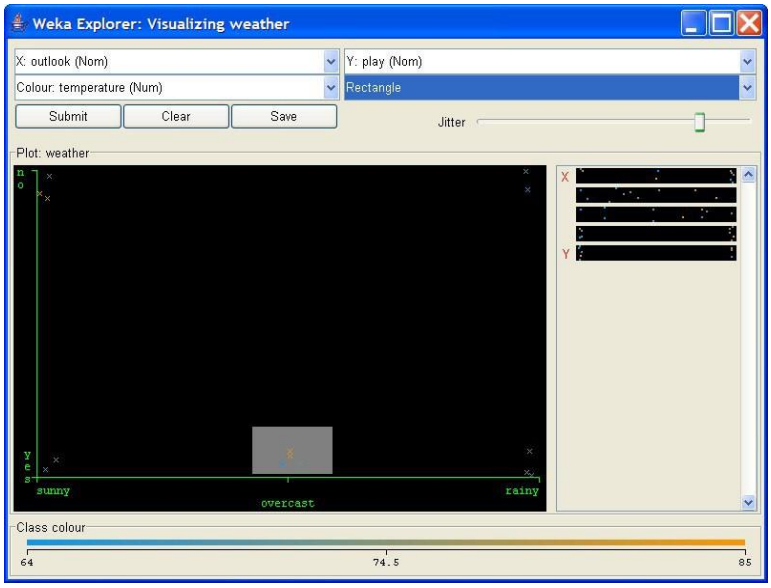
Selecting Instances

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]:

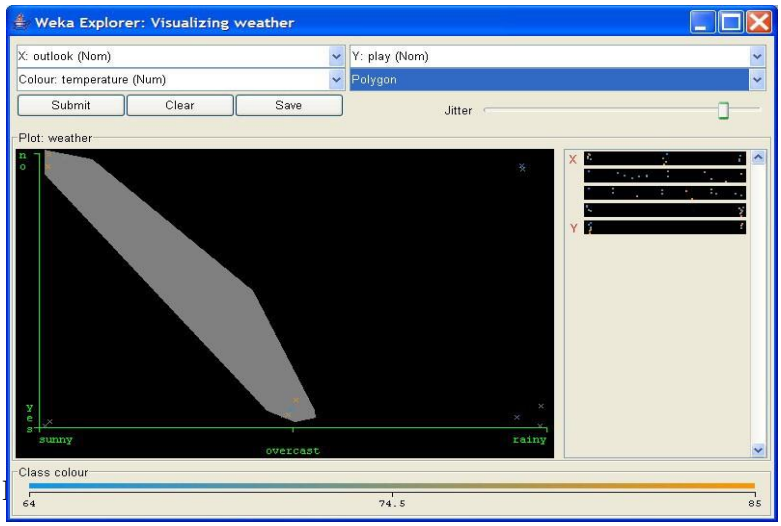
1. **Select Instance**. Click on an individual data point. It brings up a window listing

   attributes of the point. If more than one point will appear at the same location, more than one set of attributes will be shown.

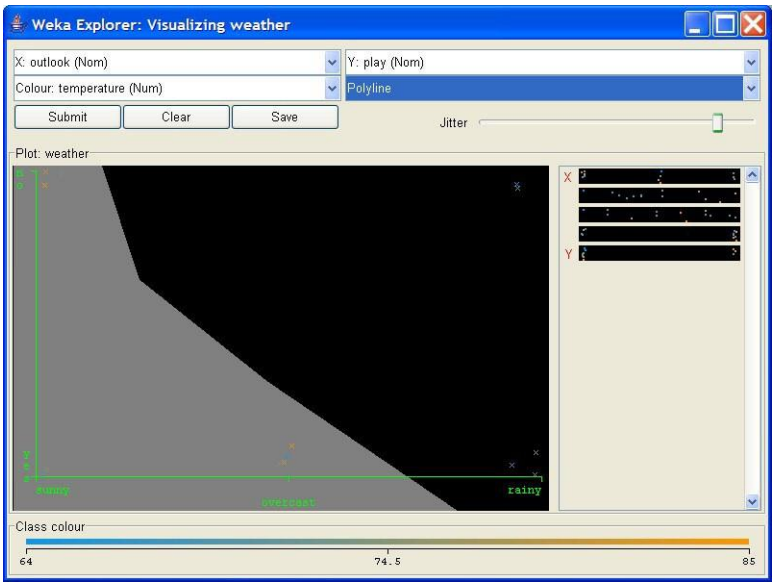
2. **Rectangle**. You can create a rectangle by dragging it around the point.



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.



B) Explore WEKA Data Mining/Machine Learning Toolkit.

Downloading and/or installation of WEKA data mining toolkit.

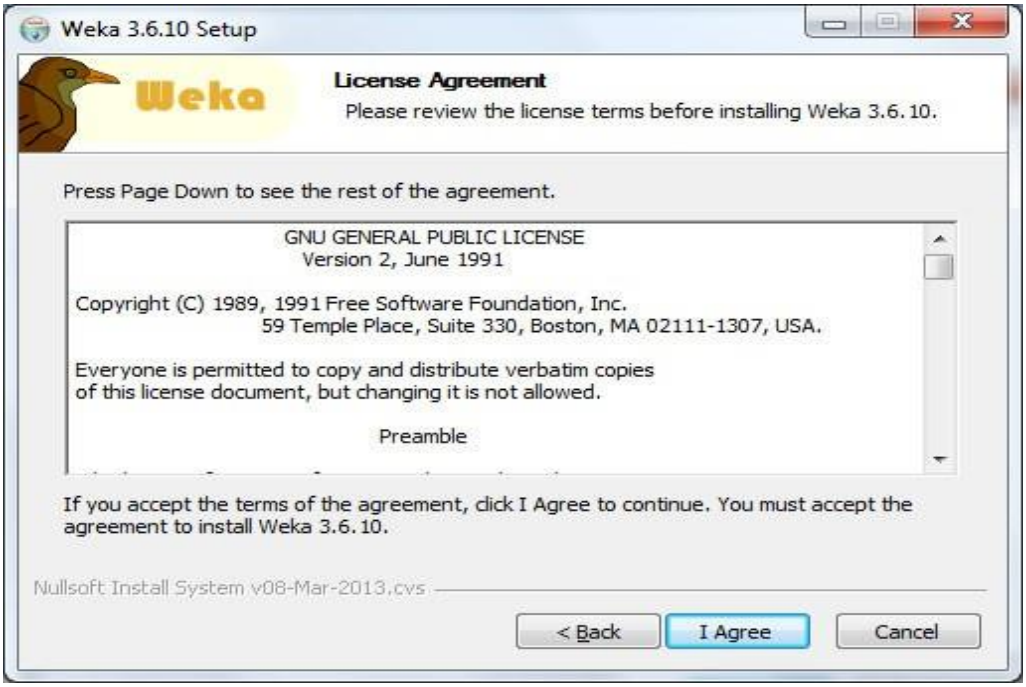
Ans:

Install Steps for WEKA a Data Mining Tool

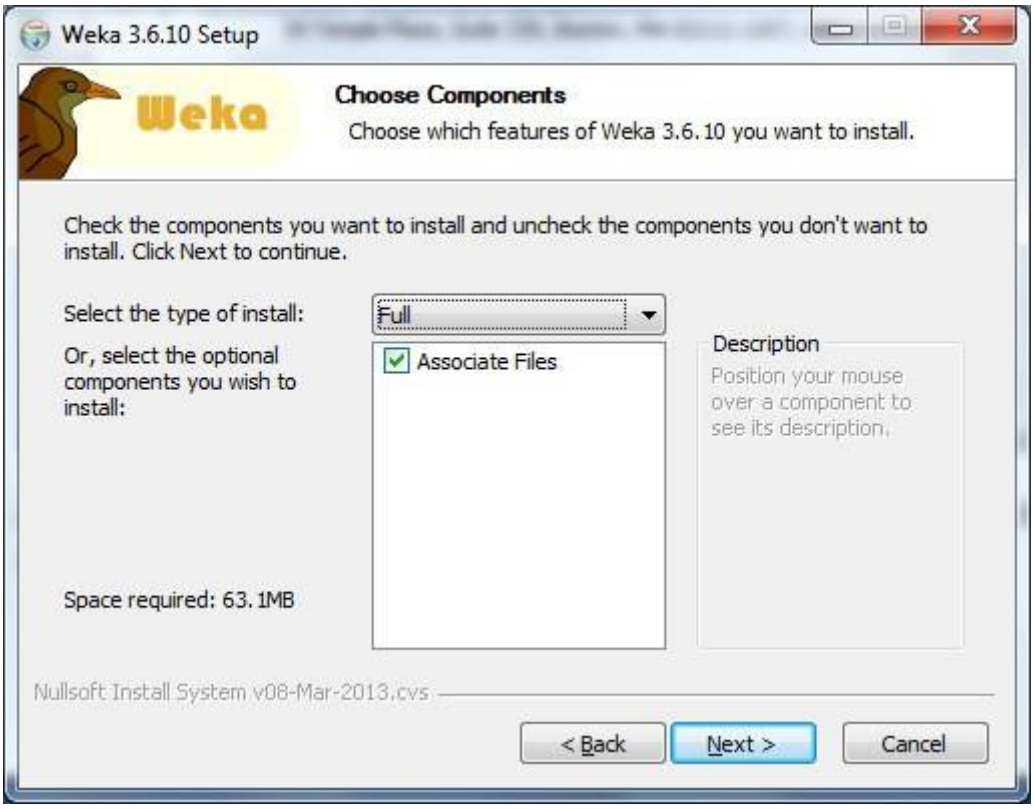
1. Download the software as your requirements from the below given link. http://www.cs.waikato.ac.nz/ml/weka/downloading.html
2. The Java is mandatory for installation of WEKA so if you have already Java on your machine then download only WEKA else download the software with JVM.
3. Then open the file location and double click on the file

4. Click Next

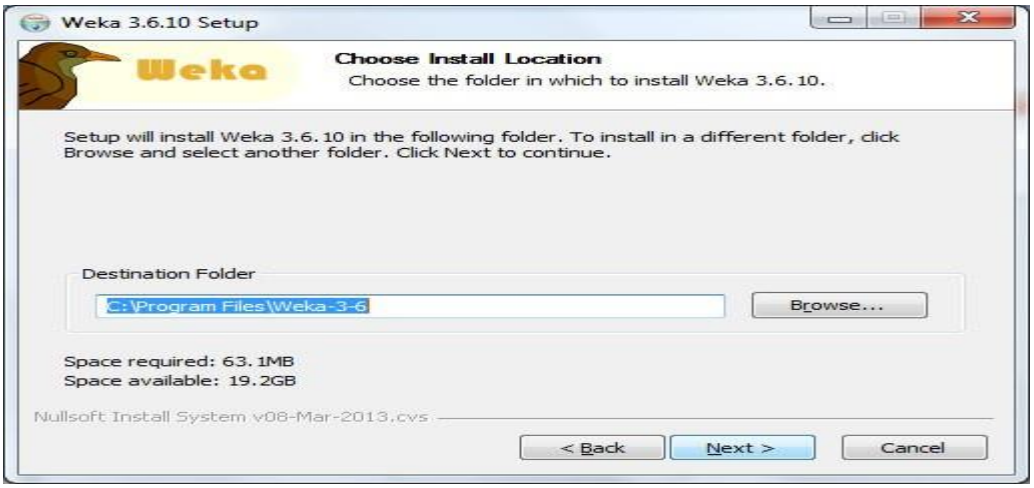
5. Click I Agree.



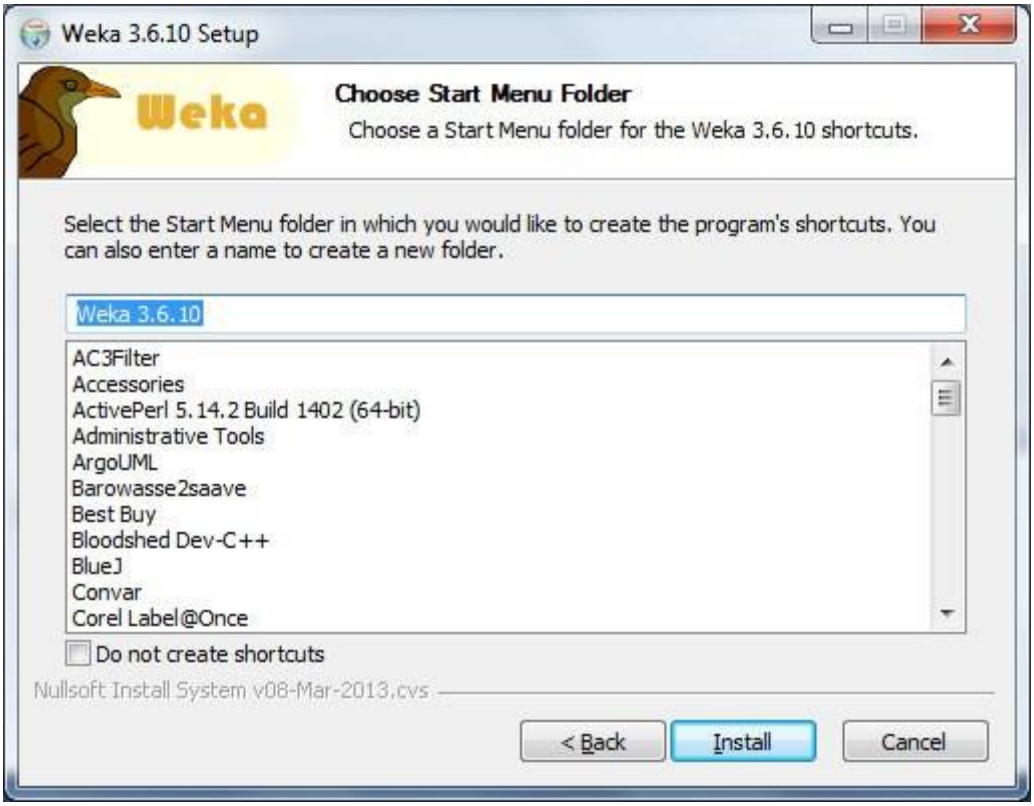
6. As your requirement do the necessary changes of settings and click Next. Full and Associate files are the recommended settings.



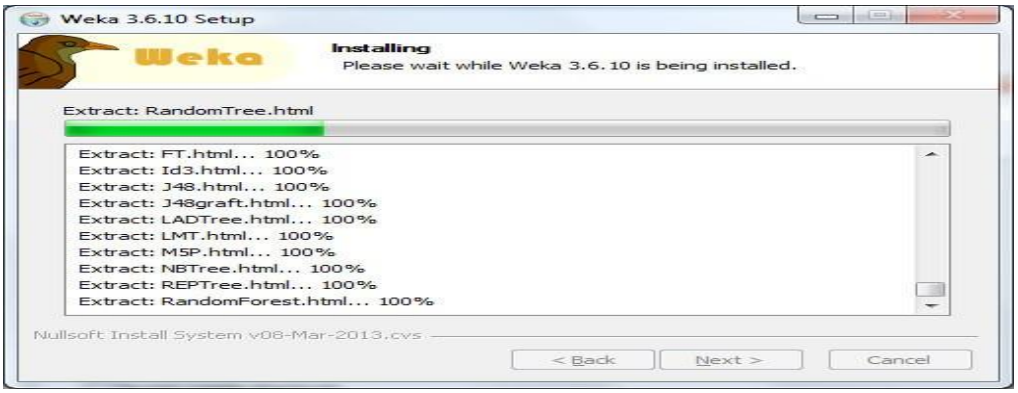
7. Change to your desire installation location.



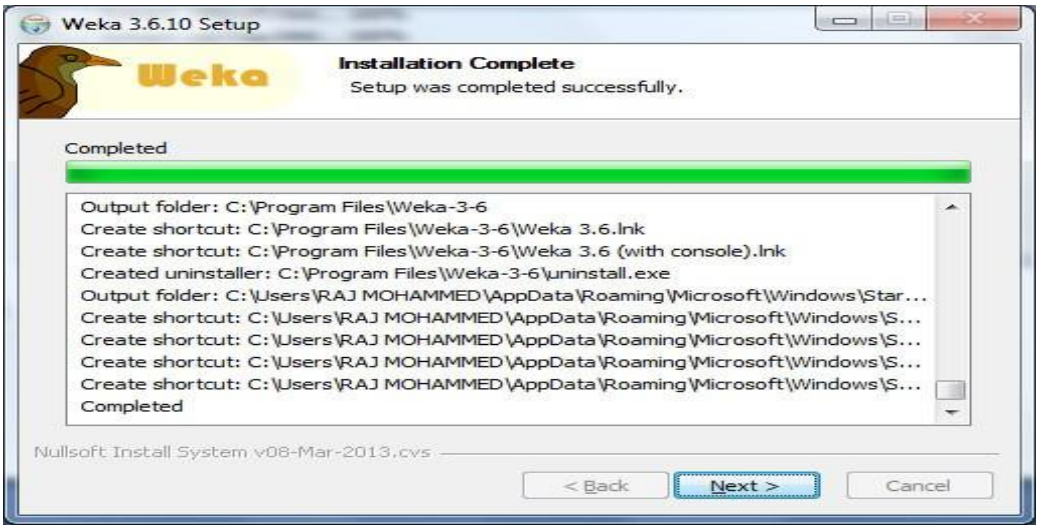
8. If you want a shortcut then check the box and click Install.



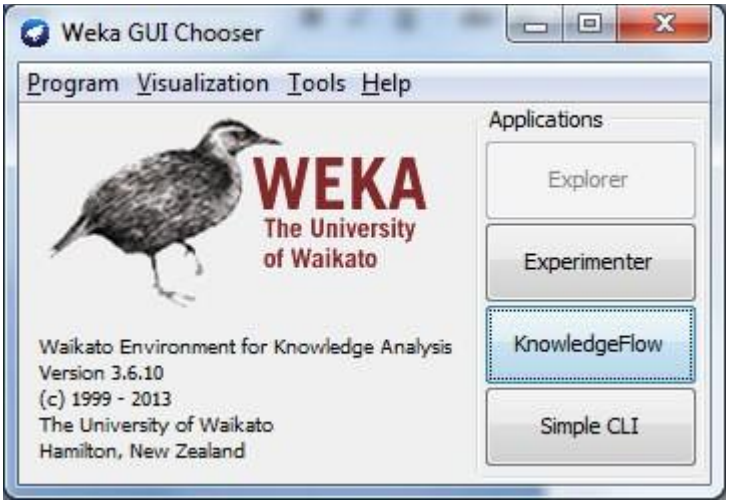
9. The Installation will start wait for a while it will finish within a minute.



10. After complete installation click on Next.



11. Hurray !!!!!!! That's all click on the Finish and take a shovel and start Mining. Best of Luck.



This is the GUI you get when started. You have 4 options Explorer, Experimenter, KnowledgeFlow and Simple CLI.

C.(ii)Understand the features of WEKA tool kit such as Explorer, Knowledge flow interface, Experimenter, command-line interface.

Ans: **WEKA**

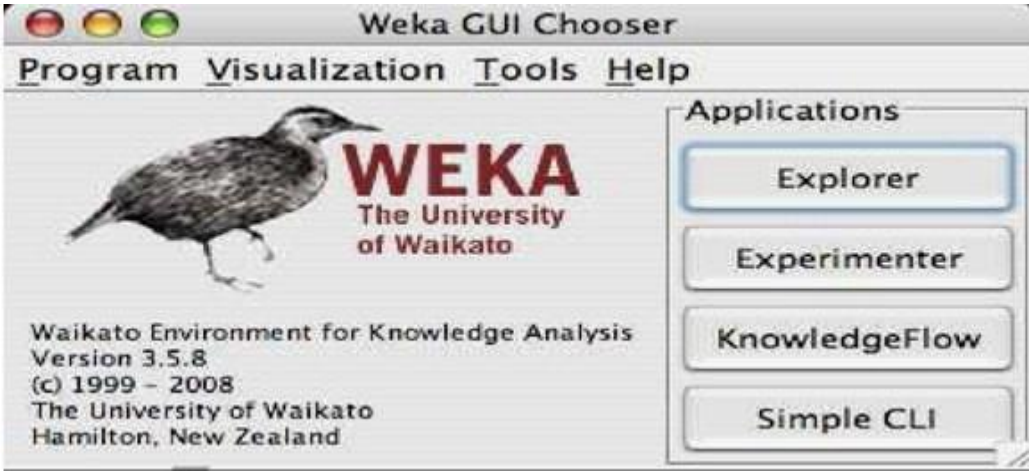
Weka is created by researchers at the university WIKATO in New Zealand. University of Waikato, Hamilton, New Zealand Alex Seewald (original Command-line primer) David Scuse (original Experimenter tutorial)

- It is java based application.
- It is collection often source, Machine Learning Algorithm.
- The routines (functions) are implemented as classes and logically arranged in packages.
- It comes with an extensive GUI Interface.
- Weka routines can be used standalone via the command line interface.

The Graphical User Interface;-

The Weka GUI Chooser (class weka.gui.GUIChooser) provides a starting point for launching Weka's main GUI applications and supporting tools. If one prefers a MDI ("multiple document interface") appearance, then this is provided by an alternative launcher called "Main"

(class weka.gui.Main). The GUI Chooser consists of four buttons—one for each of the four major Weka applications—and four menus.



The buttons can be used to start the following applications:

- **Explorer An environment** for exploring data with WEKA (the rest of this Documentation deals with this application in more detail).
- **Experimenter** An environment for performing experiments and conducting statistical tests between learning schemes.

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

- **SimpleCLI Provides** a simple command-line interface that allows direct execution of WEKA commands for operating systems that do not provide their own command line interface.

1. Explorer

The Graphical user interface

1.1 Section Tabs

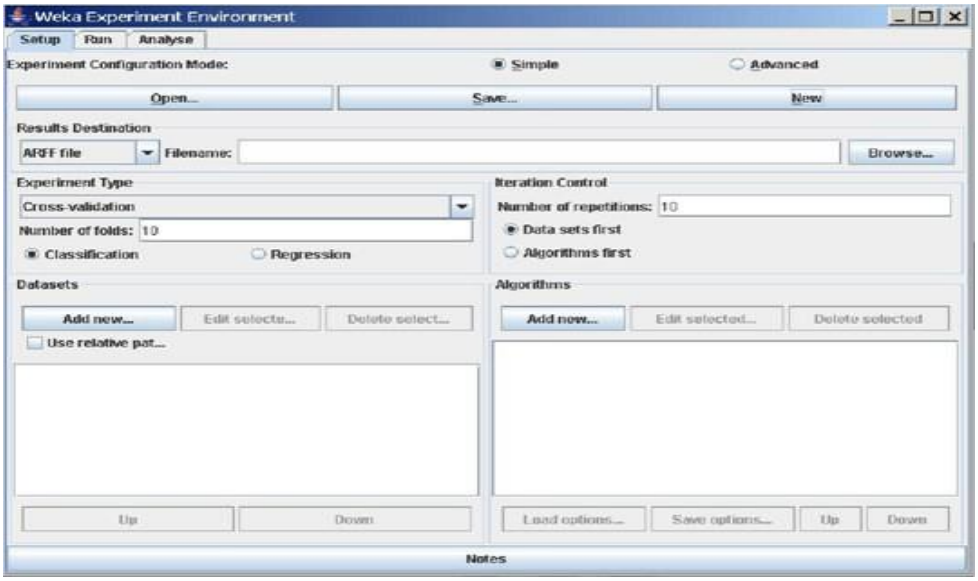
At the very top of the window, just below the title bar, is a row of tabs. When the Explorer is first started only the first tab is active; the others are grayed out. This is because it is necessary to open (and potentially pre-process) a data set before starting to explore the data.
The tabs are as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

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

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

2.Weka Experimenter:-

The Weka Experiment Environment enables the user to create, run, modify, and analyze experiments in a more convenient manner than is possible when processing the schemes individually. For example, the user can create an experiment that runs several schemes against a series of datasets and then analyze the results to determine if one of the schemes is (statistically) better than the other schemes.



The Experiment Environment can be run from the command line using the Simple CLI. For example, the following commands could be typed into the CLI to run the OneR scheme on the Iris dataset using a basic train and test process. (Note that the commands would be typed on one line into the CLI.) While commands can be typed directly into the CLI, this technique is not particularly convenient and the experiments are not easy to modify. The Experimenter comes in two flavors', either with a simple interface that provides most of the functionality one needs for experiments, or with an interface **with full access to the Experimenter's capabilities. You can** choose between those two with the Experiment Configuration Mode radio buttons:

- Simple
- Advanced

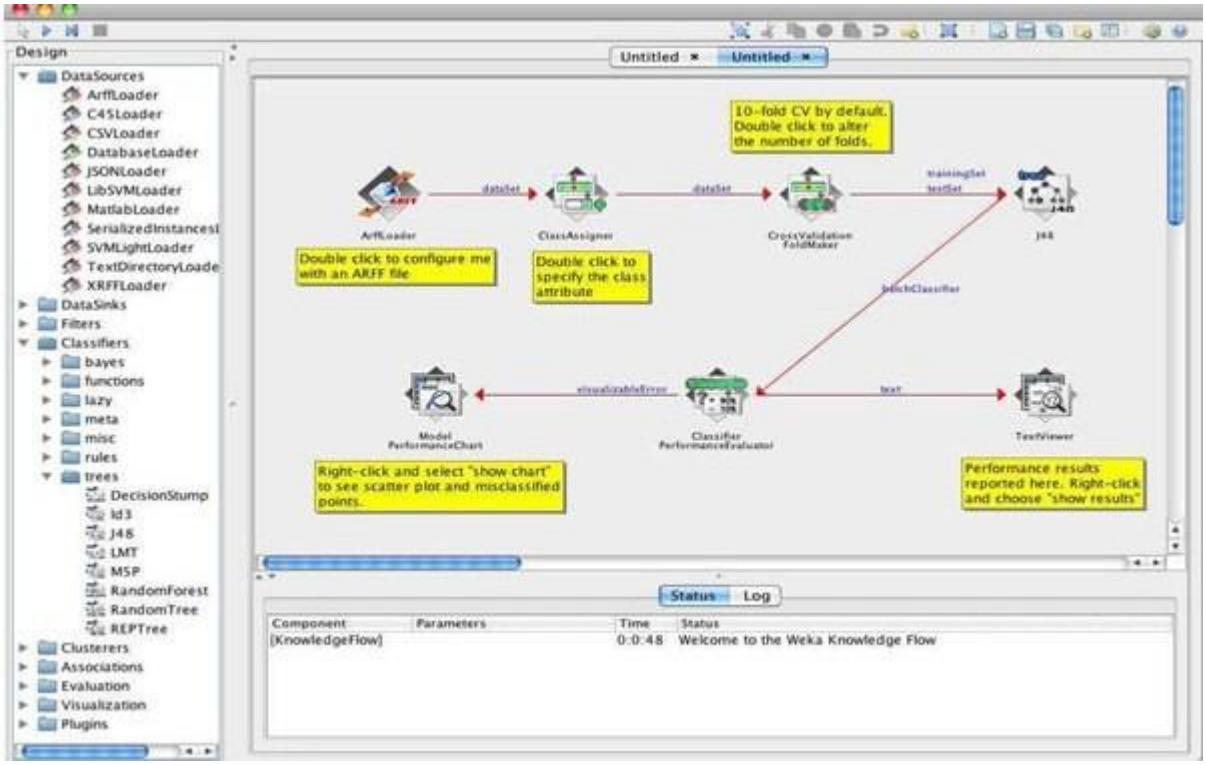
Both setups allow you to setup standard experiments, that are run locally on a single machine, or remote experiments, which are distributed between several hosts. The distribution of experiments cuts down the time the experiments will take until completion, but on the other hand the setup takes more time. The next section covers the standard experiments (both, simple and advanced), followed by the remote experiments and finally the analyzing of the results.

3. Knowledge Flow

Introduction

The Knowledge Flow provides an alternative to the Explorer as a graphical front end to **WEKA's core algorithms.**

The Knowledge Flow presents a data-flow inspired interface to WEKA. The user can select WEKA components from a palette, place them on a layout canvas and connect them together in order to form a knowledge flow for processing and analyzing data. At present, all of **WEKA's classifiers, filters, clusterers, associators, loaders and savers a**re available in the Knowledge Flow along with some extra tools.



The Knowledge Flow can handle data either incrementally or in batches (the Explorer handles batch data only). Of course learning from data incremen- tally requires a classifier that can

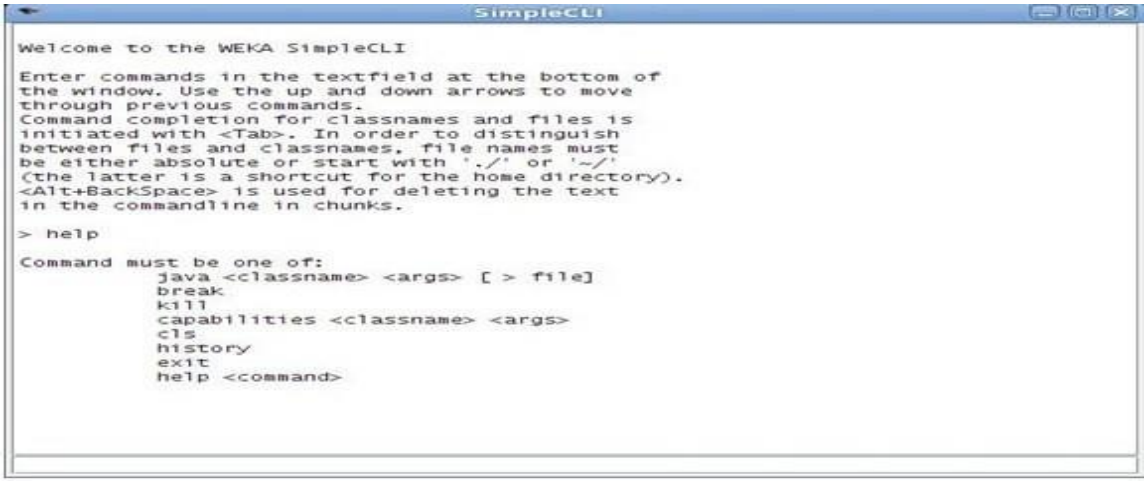
be updated on an instance by instance basis. Currently in WEKA there are ten classifiers that can handle data incrementally.

The Knowledge Flow offers the following features:

- **Intuitive** data flow style layout.
- **Process** data in batches or incrementally.
- **Process multiple batches** or streams in parallel (each separate flow executes in its own thread) .
- **Process multiple streams sequentially** via a user-specified order of execution.
- **Chain filters** together.
- **View models** produced by classifiers for each fold in a cross validation.
- **Visualize performance** of incremental classifiers during processing (scrolling plots of classification accuracy, RMS error, predictions etc.).
- **Plugin "perspectives" that add major new functionality (e.g. 3D data** visualization, time series forecasting environment etc.).

4.Simple CLI

The Simple CLI provides full access to all Weka classes, i.e., classifiers, filters, clusterers, etc., but without the hassle of the CLASSPATH (it facilitates the one, with which Weka was started). It offers a simple Weka shell with separated command line and output.



Commands

The following commands are available in the Simple CLI:

- Java <classname> [<args>]

  Invokes a java class with the given arguments (if any).

- Break

  Stops the current thread, e.g., a running classifier, in a friendly manner kill stops the current thread in an unfriendly fashion.

- Cls

  Clears the output area

- Capabilities <classname> [<args>]

  Lists the capabilities of the specified class, e.g., for a classifier with its.

- option:

  Capabilities weka.classifiers.meta.Bagging -W weka.classifiers.trees.Id3

- exit

  Exits the Simple CLI

- help [<command>]

  Provides an overview of the available commands if without a command name as argument, otherwise more help on the specified command

Invocation

In order to invoke a Weka class, one has only to prefix the class with "java". This command tells the Simple CLI to load a class and execute it with any given parameters. E.g., the J48 classifier can be invoked on the iris dataset with the following command:

java weka.classifiers.trees.J48 -t c:/temp/iris.arff

This results in the following output:

Command redirection

Starting with this version of Weka one can perform a basic redirection: java weka.classifiers.trees.J48 -t test.arff > j48.txt

Note: the > must be preceded and followed by a space, otherwise it is not recognized as redirection, but part of another parameter.

Command completion

Commands starting with java support completion for classnames and filenames via Tab (Alt+BackSpace deletes parts of the command again). In case that there are several matches, Weka lists all possible matches.

- Package Name Completion java weka.cl<Tab>

  Results in the following output of possible matches of

package names: Possible matches:

weka.classifiers
weka.clusterers

- Classname completion

java weka.classifiers.meta.A<Tab> lists the following classes

Possible matches:

weka.classifiers.meta.AdaBoostM1
weka.classifiers.meta.AdditiveRegression
weka.classifiers.meta.AttributeSelectedClassifier

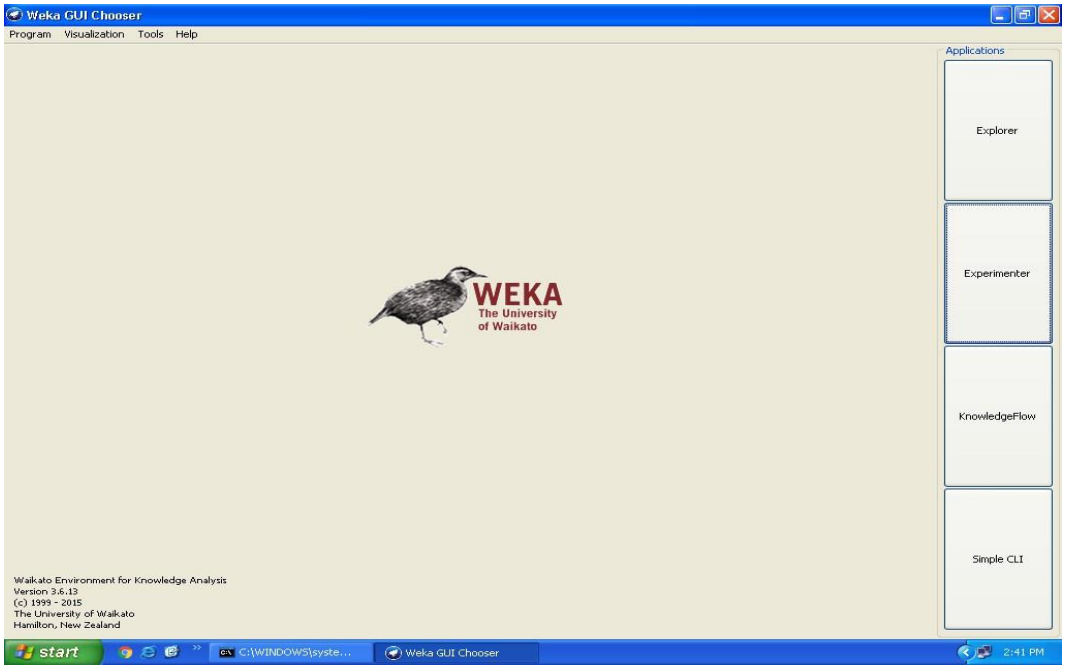
- Filename Completion

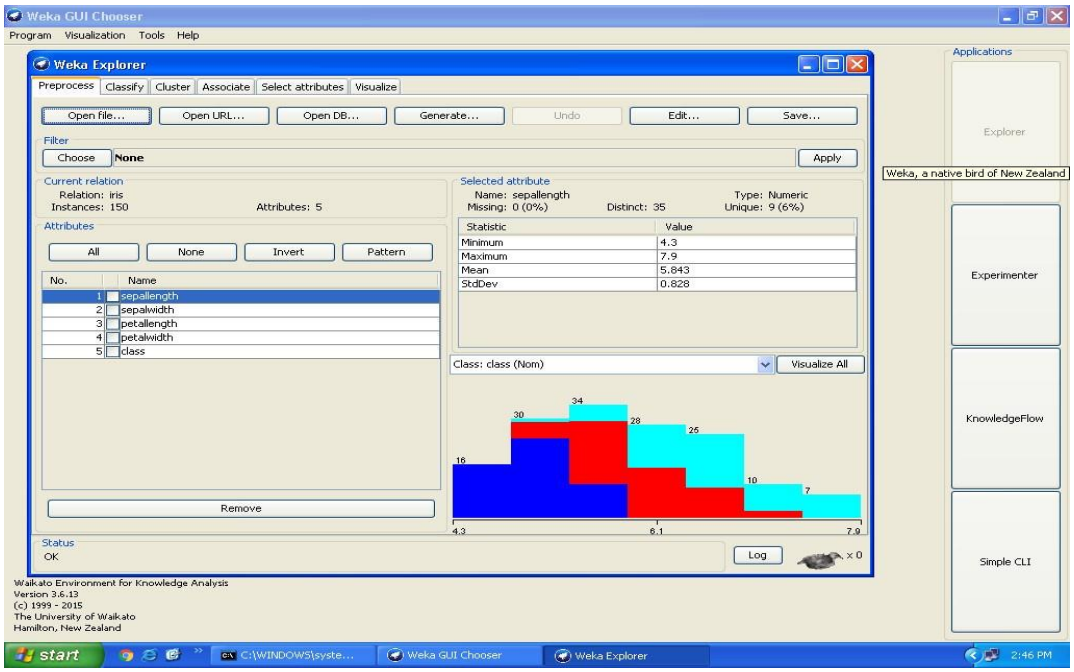
In order for Weka to determine whether a the string under the cursor is a classname or a filename, filenames need to be absolute (Unix/Linx: /some/path/file;Windows: C:\Some\Path\file) or relative and starting with a dot (Unix/Linux:./some/other/path/file; Windows: .\Some\Other\Path\file).

D.(iii)Navigate the options available in the WEKA(ex.select attributes panel,preprocess panel,classify panel,cluster panel,associate panel and visualize)

Ans: Steps for identify options in WEKA

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on Preprocessing tab button.
4. Click on open file button.
5. Choose WEKA folder in C drive.
6. Select and Click on data option button.
7. Choose iris data set and open file.
8. All tabs available in WEKA home page.


Weka GUI Chooser
Program Visualization Tools Help
Applications
Explorer
Experimenter
KnowledgeFlow
Simple CLI
WEKA
The University of Waikato
Waikato Environment for Knowledge Analysis
Version 3.6.13
(c) 1999 - 2015
The University of Waikato
Hamilton, New Zealand
start
C:\WINDOWS\syste...
Weka GUI Chooser
2:41 PM


Weka GUI Chooser
Program Visualization Tools Help
Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose None Apply
Current relation
Relation: iris
Instances: 150
Attributes: 5
Selected attribute
Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)
Attributes
All None Invert Pattern
No. Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Statistic Value
Minimum 4.3
Maximum 7.9
Mean 5.843
StdDev 0.828
Class: class (Nom)
Visualize All
Remove
Status
OK
Log
x 0
Applications
Explorer
Weka, a native bird of New Zealand
Experimenter
KnowledgeFlow
Simple CLI
Waikato Environment for Knowledge Analysis
Version 3.6.13
(c) 1999 - 2015
The University of Waikato
Hamilton, New Zealand
start
C:\WINDOWS\syste...
Weka GUI Chooser
Weka Explorer
2:46 PM

Study the ARFF file format

Ans: **ARFF File Format**

An ARFF (= Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

ARFF files are not the only format one can load, but all files that can be converted with Weka's "core converters". The following formats are currently supported:

- ARFF (+ compressed)
- C4.5
- CSV
- libsvm
- binary serialized instances
- XRFF (+ compressed)

Overview

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

An example header on the standard IRIS dataset looks like this:

1. Title: Iris Plants Database

2. Sources:

(a) Creator: R.A. Fisher
(b) Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)

(c) Date: July, 1988

@RELATION iris

@ATTRIBUTE sepal length NUMERIC
@ATTRIBUTE sepal width NUMERIC
@ATTRIBUTE petal length NUMERIC
@ATTRIBUTE petal width NUMERIC
@ATTRIBUTE class {Iris-setosa, Iris-versicolor, Iris-irginica} The Data of the ARFF file looks like the following:

@DATA

5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa

Lines that begin with a % are comments.
The @RELATION, @ATTRIBUTE and @DATA declarations are case insensitive.

The ARFF Header Section

The ARFF Header section of the file contains the relation declaration and at• tribute declarations.

The @relation Declaration

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

The @attribute Declarations

Attribute declarations take the form of an ordered sequence of @attribute statements. Each attribute in the data set has its own @attribute statement which uniquely defines the name **of that attribute and it's data type. The order** the attributes are declared indicates the column position in the data section of the file. For example, if an attribute is the third one declared then Weka expects that all that attributes values will be found in the third comma delimited column.

The format for the @attribute statement is:

@attribute <attribute-name> <datatype>

where the <attribute-name> must start with an alphabetic character. If spaces are to be included in the name then the entire name must be quoted.

The <datatype> can be any of the four types supported by Weka:

- numeric
- integer is treated as numeric
- real is treated as numeric
- <nominal-specification>
- string
- date [<date-format>]
- relational for multi-instance data (for future use)

where <nominal-specification> and <date-format> are defined below. The keywords numeric, real, integer, string and date are case insensitive.

Numeric attributes

Numeric attributes can be real or integer numbers.

Nominal attributes

Nominal values are defined by providing an <nominal-specification> listing the possible values: <nominal-name1>, <nominal-name2>, <nominal-name3>,
For example, the class value of the Iris dataset can be defined as follows: @ATTRIBUTE class {Iris-setosa,Iris-versicolor,Iris-virginica} Values that contain spaces must be quoted.

String attributes

String attributes allow us to create attributes containing arbitrary textual values. This is very useful in text-mining applications, as we can create datasets with string attributes, then write Weka Filters to manipulate strings (like String- ToWordVectorFilter). String attributes are declared as follows:

@ATTRIBUTE LCC string

Date attributes

Date attribute declarations take the form: @attribute <name> date [<date-format>] where <name> is the name for the attribute and <date-format> is an optional string specifying how date values should be parsed and printed (this is the same format used by SimpleDateFormat). The default format string accepts the ISO-8601 combined date and time format: yyyy-MM-**dd'T'HH:mm:ss.** Dates must be specified in the data section as the corresponding string representations of the date/time (see example below).

Relational attributes

Relational attribute declarations take the form: @attribute <name> relational
<further attribute definitions> @end <name>
For the multi-instance dataset MUSK1 the definition would look like thi**s (”...”** denotes an omission):
@attribute molecule_name {MUSK-jf78,...,NON-MUSK-199} @attribute bag relational

@attribute f1 numeric
...
@attribute f166 numeric @end bag
@attribute class {0,1}

The ARFF Data Section

The ARFF Data section of the file contains the data declaration line and the actual instance lines.

The @data Declaration

The @data declaration is a single line denoting the start of the data segment in the file. The format is:

@data

The instance data

Each instance is represented on a single line, with carriage returns denoting the end of the instance. A percent sign (%) introduces a comment, which continues to the end of the line.

Attribute values for each instance are delimited by commas. They must appear in the order that they were declared in the header section (i.e. the data corresponding to the nth @attribute declaration is always the nth field of the attribute).

Missing values are represented by a single question mark, as in:

@data 4.4,?,1.5,?,Iris-setosa

Values of string and nominal attributes are case sensitive, and any that contain space or the comment-delimiter character % must be quoted. (The code suggests that double-quotes are

acceptable and that a backslash will escape individual characters.)string
An example follows: @relation LCCvsLCSH @attribute LCC string @attribute LCSH@data

AG5, 'Encyclopedias and dictionaries.;Twentieth century.' AS262, 'Science **-- Soviet Union --** History.' AE5, 'Encyclopedias and dictionaries.'

AS281, 'Astronomy, Assyro-Babylonian.;Moon -- **Phases.'**
AS28**1, 'Astronomy, Assyro**-Babylonian.;Moon -- **Tables.'**

Dates must be specified in the data section using the string representation specified in the attribute declaration.

For example:
@RELATION Timestamps
@ATTRIBUTE timestamp DATE "yyyy-MM-dd HH:mm:ss" @DATA

"2001-04-03 12:12:12"
"2001-05-03 12:59:55"

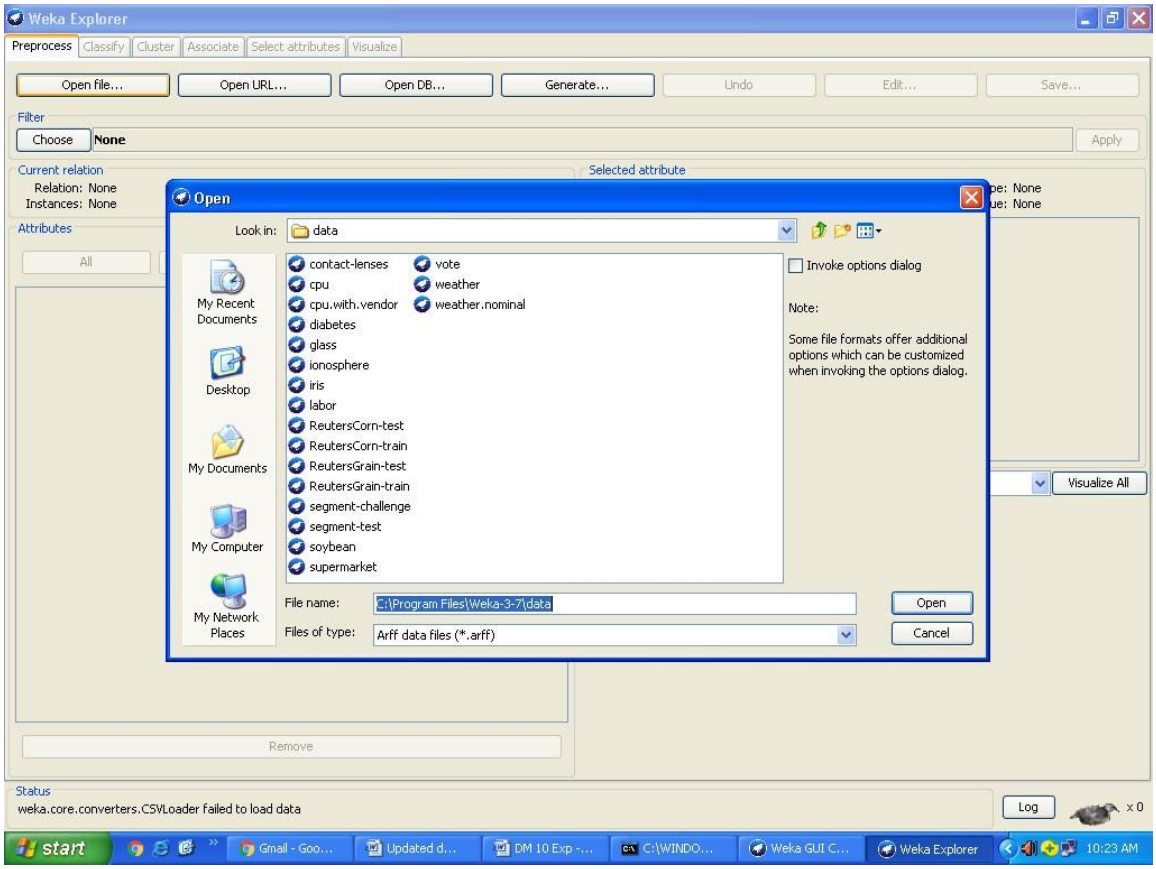
Relational data must be enclosed within double quotes ". For example an instance of the MUSK1 **dataset ("..." denotes an omission):**

MUSK-188,"42,...,30",1

Explore the available data sets in WEKA.

Ans: Steps for identifying data sets in WEKA

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on open file button.
4. Choose WEKA folder in C drive.
5. Select and Click on data option button.



Sample Weka Data Sets

Below are some sample WEKA data sets, in arff format.

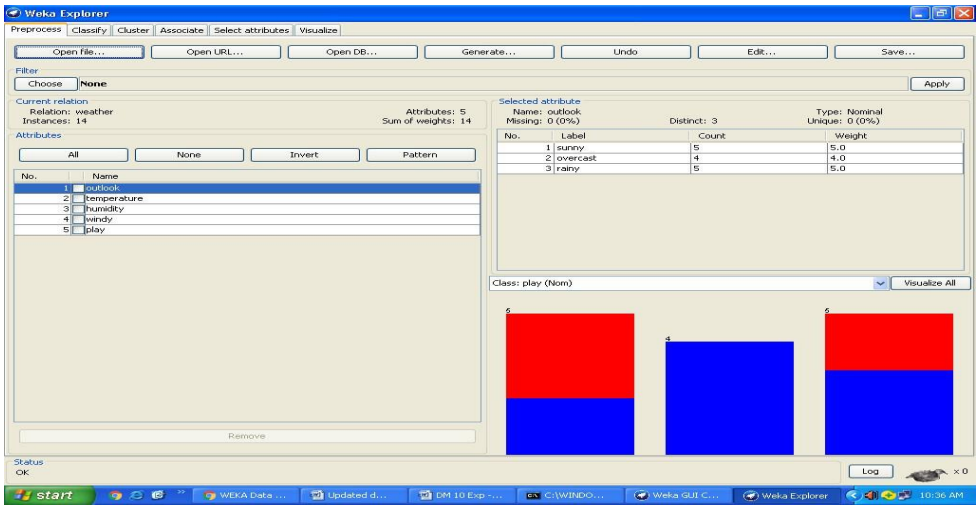
- contact-lens.arff
- cpu.arff
- cpu.with-vendor.arff
- diabetes.arff
- glass.arff
- ionospehre.arff
- iris.arff
- labor.arff
- ReutersCorn-train.arff
- ReutersCorn-test.arff

- ReutersGrain-train.arff
- ReutersGrain-test.arff
- segment-challenge.arff
- segment-test.arff
- soybean.arff
- supermarket.arff
- vote.arff
- weather.arff
- weather.nominal.arff

Load a data set (ex.Weather dataset,Iris dataset,etc.)

Ans: Steps for load the Weather data set.

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on open file button.
4. Choose WEKA folder in C drive.
5. Select and Click on data option button.
6. Choose Weather.arff file and Open the file.



EXERCISE-1

1. Write Steps for load the Iris data set.

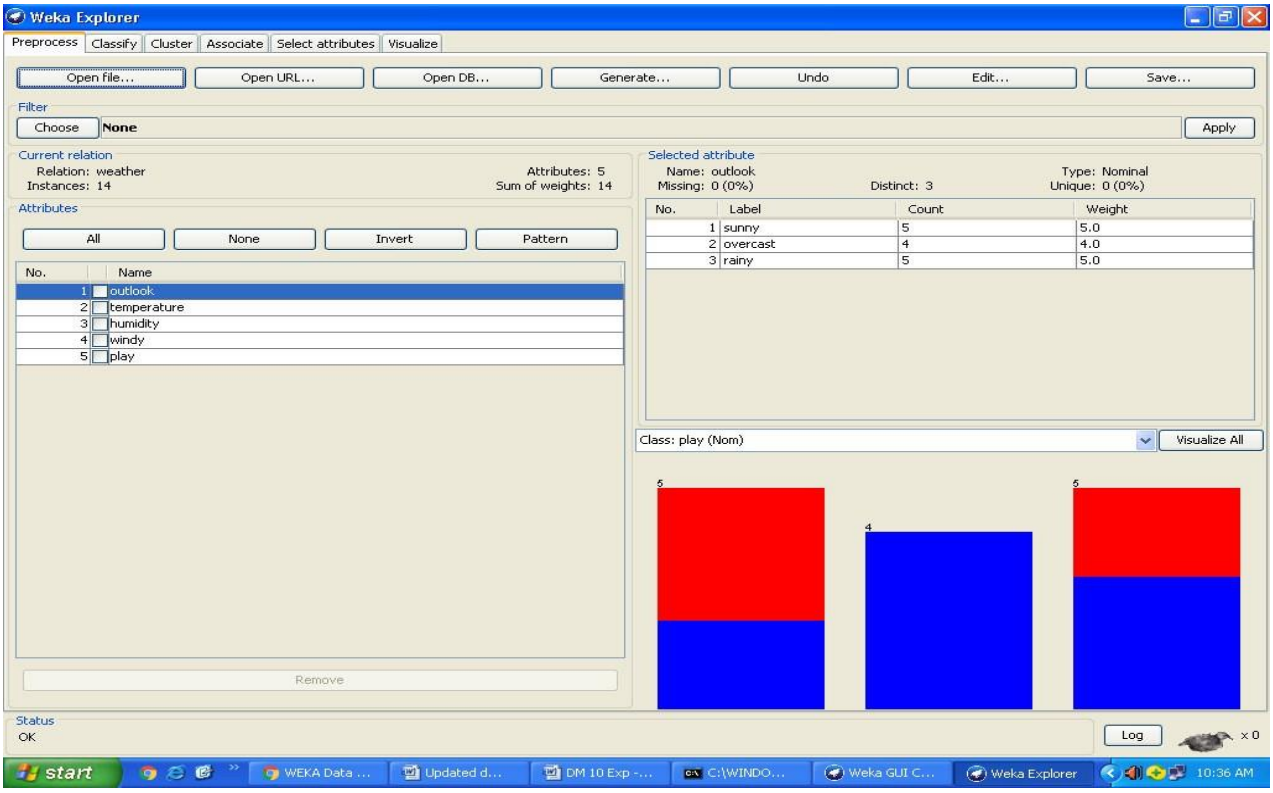
Load each dataset and observe the following:

List attribute names and types

Eg: dataset-Weather.arff

List out the attribute names:

1. outlook
2. temperature
3. humidity
4. windy
5. play



EXERCISE 2:

List attribute names and types of Dataset SuperMarket.

Number of records in each dataset.

Ans: @relation weather.symbolic

@attribute outlook {sunny, overcast, rainy}
@attribute temperature {hot, mild, cool}
@attribute humidity {high, normal} @attribute
windy {TRUE, FALSE} @attribute play {yes,
no}
@data sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no

Identify the class attribute (if any)

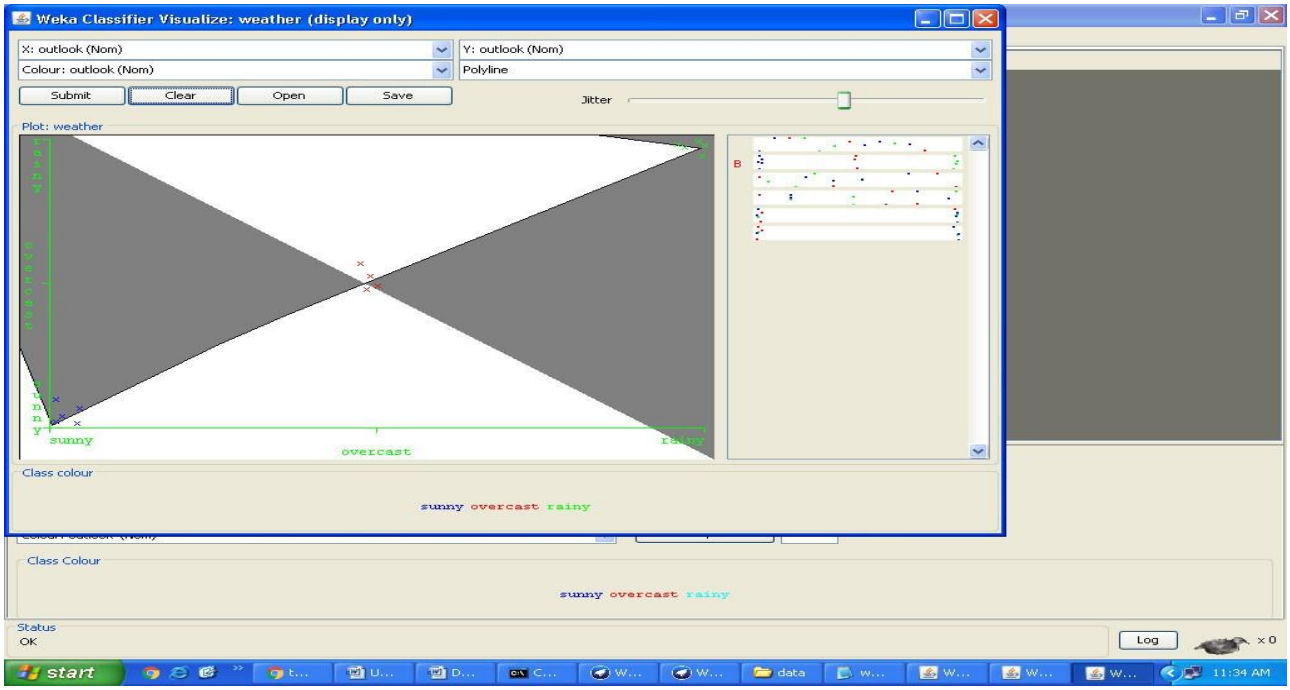
Ans: class attributes

1. sunny
2. overcast
3. rainy

Plot Histogram

Steps for identify the plot histogram

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on Visualize button.
4. Click on right click button.
5. Select and Click on polyline option button.



EXERCISE 3: Plot Histogram of Different Datasets

Eg: IRIS,Contactlense etc..

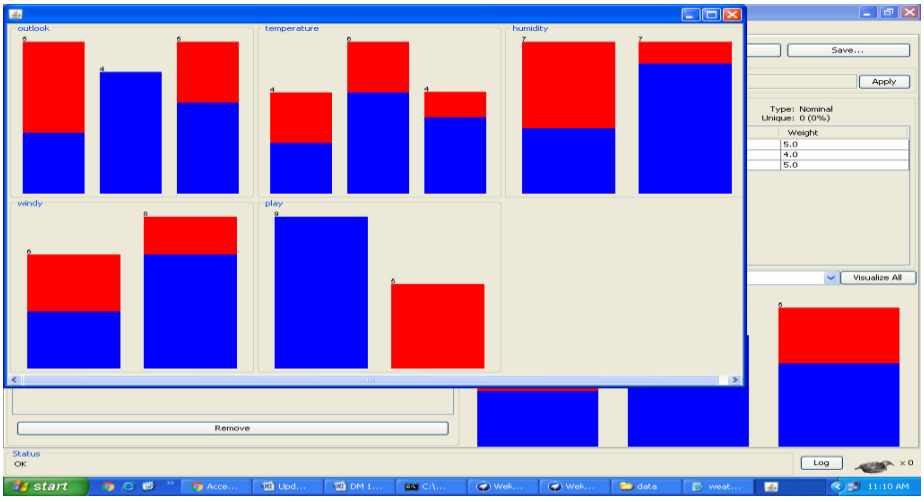
Determine the number of records for each class

Ans: @relation weather.symbolic
@data

sunny,hot,high,FALSE,no
sunny,hot,high,TRUE,no
overcast,hot,high,FALSE,yes
rainy,mild,high,FALSE,yes
rainy,cool,normal,FALSE,yes
rainy,cool,normal,TRUE,no
overcast,cool,normal,TRUE,yes
sunny,mild,high,FALSE,no
sunny,cool,normal,FALSE,yes
rainy,mild,normal,FALSE,yes
sunny,mild,normal,TRUE,yes
overcast,mild,high,TRUE,yes
overcast,hot,normal,FALSE,yes
rainy,mild,high,TRUE,no

Visualize the data in various dimensions

Click on Visualize All button in WEKA Explorer.



Viva voice questions:

1. What is data warehouse?

A data warehouse is a electronic storage of an Organization's historical data for the purpose of reporting, analysis and data mining or knowledge discovery.

2. What is the benefits of data warehouse?

A data warehouse helps to integrate data and store them historically so that we can analyze different aspects of business including, performance analysis, trend, prediction etc. over a given time frame and use the result of our analysis to improve the efficiency of business processes.

3. What is Fact?

A fact is something that is quantifiable (Or measurable). Facts are typically (but not always) numerical values that can be aggregated.

SIGNATURE OF FACULTY

WEEK 2-

Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

A. Explore various options in Weka for Preprocessing data and apply (like Discretization Filters, Resample filter, etc.) n each dataset.

Ans:

Preprocess Tab

1. Loading Data

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

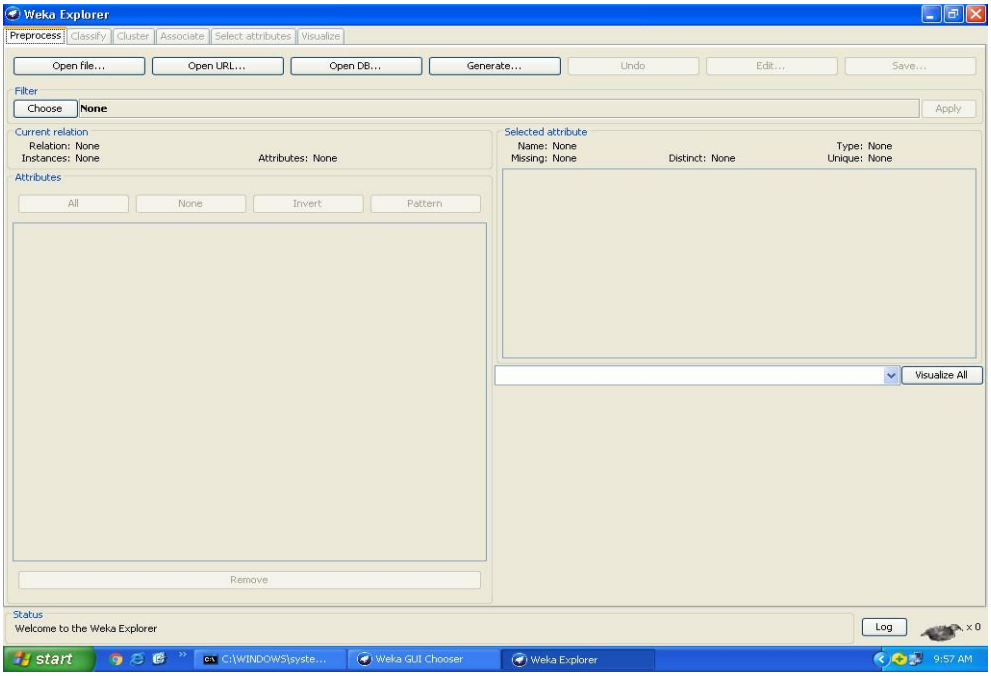
1. **Open file....** Brings up a dialog box allowing you to browse for the data file on the local file system.

2. **Open URL....** Asks for a Uniform Resource Locator address for where the data is stored.

3. **Open DB ....**Reads data from a database. (Note that to make this work you might have to edit the file in weka/experiment/DatabaseUtils.props.)

4. **Generate ....**Enables you to generate artificial data from a variety of Data Generators. Using the Open file ...button you can read files in a variety of formats: **WEKA's ARFF format, CSV**

format, C4.5 format, or serialized Instances format. ARFF files typically have a .arff extension, CSV files a .csv extension, C4.5 files a .data and .names extension, and serialized Instances objects a .bsi extension.



Current Relation: Once some data has been loaded, the Preprocess panel shows a variety of in**formation. The Current relation box (the "current relation" is the** currently loaded data, which can be interpreted as a single relational table in database terminology) has three entries:

1. Relation. The name of the relation, as given in the file it was loaded from. Filters (described below) modify the name of a relation.

2. Instances. The number of instances (data points/records) in the data.

3. Attributes. The number of attributes (features) in the data.

Working With Attributes

Below the Current relation box is a box titled Attributes. There are four buttons, and beneath them is a list of the attributes in the current relation.

The list has three columns:

1. No.. A number that identifies the attribute in the order they are specified in the data file.

2. Selection tick boxes. These allow you select which attributes are present in the relation.
3. Name. The name of the attribute, as it was declared in the data file. When you click on different rows in the list of attributes, the fields change in the box to the right titled Selected attribute.

This box displays the characteristics of the currently highlighted attribute in the list:

1. Name. The name of the attribute, the same as that given in the attribute list.

2. Type. The type of attribute, most commonly Nominal or Numeric.

3. Missing. The number (and percentage) of instances in the data for which this attribute is missing (unspecified).
4. Distinct. The number of different values that the data contains for this attribute.

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

Below these statistics is a list showing more information about the values stored in this attribute, which differ depending on its type. If the attribute is nominal, the list consists of each possible value for the attribute along with the number of instances that have that value. If the attribute is numeric, the list gives four statistics describing the distribution of values in the data— the minimum, maximum, mean and standard deviation. And below these statistics there is a coloured histogram, colour-coded according to the attribute chosen as the Class using the box above the histogram. (This box will bring up a drop-down list of available selections when clicked.) Note that only nominal Class attributes will result in a colour-coding. Finally, after pressing the Visualize All button, histograms for all the attributes in the data are shown in a separate window.Returning to the attribute list, to begin with all the tick boxes are unticked.

They can be toggled on/off by clicking on them individually. The four buttons above can also be used to change the selection:

PREPROCESSING

1. **All.** All boxes are ticked.
2. **None.** All boxes are cleared (unticked).
3. **Invert.** Boxes that are ticked become unticked and vice versa.

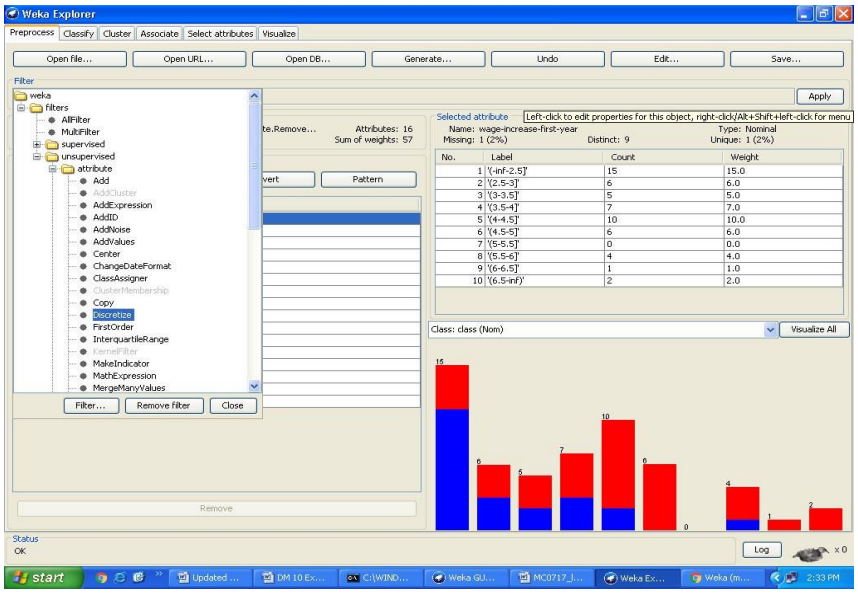
4. **Pattern.** Enables the user to select attributes based on a Perl 5 Regular Expression. E.g., .* id selects all attributes which name ends with id.

Once the desired attributes have been selected, they can be removed by clicking the Remove button below the list of attributes. Note that this can be undone by clicking the Undo button, which is located next to the Edit button in the top-right corner of the Preprocess panel.

Working with Filters:-

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

Clicking on this box with the left mouse button brings up a GenericObjectEditor dialog box. A click with the right mouse button (or Alt+Shift+left click) brings up a menu where you can choose, either to display the properties in a GenericObjectEditor dialog box, or to copy the current setup string to the clipboard.



The GenericObjectEditor Dialog Box

The GenericObjectEditor dialog box lets you configure a filter. The same kind of dialog box is used to configure other objects, such as classifiers and clusterers

(see below). The fields in the window reflect the available options.

Right-clicking (or Alt+Shift+Left-Click) on such a field will bring up a popup menu, listing the following options:

1. Show properties... has the same effect as left-clicking on the field, i.e., a dialog appears allowing you to alter the settings.

2. Copy configuration to clipboard copies the currently displayed configuration string to the **system's clipboar**d and therefore can be used anywhere else in WEKA or in the console. This is rather handy if you have to setup complicated, nested schemes.

3. Enter configuration... is the "receiving" end for configurations that got copied to the clipboard

earlier on. In this dialog you can enter a class name followed by options (if the class supports these). This also allows you to transfer a filter setting from the Preprocess panel to a Filtered Classifier used in the Classify panel.

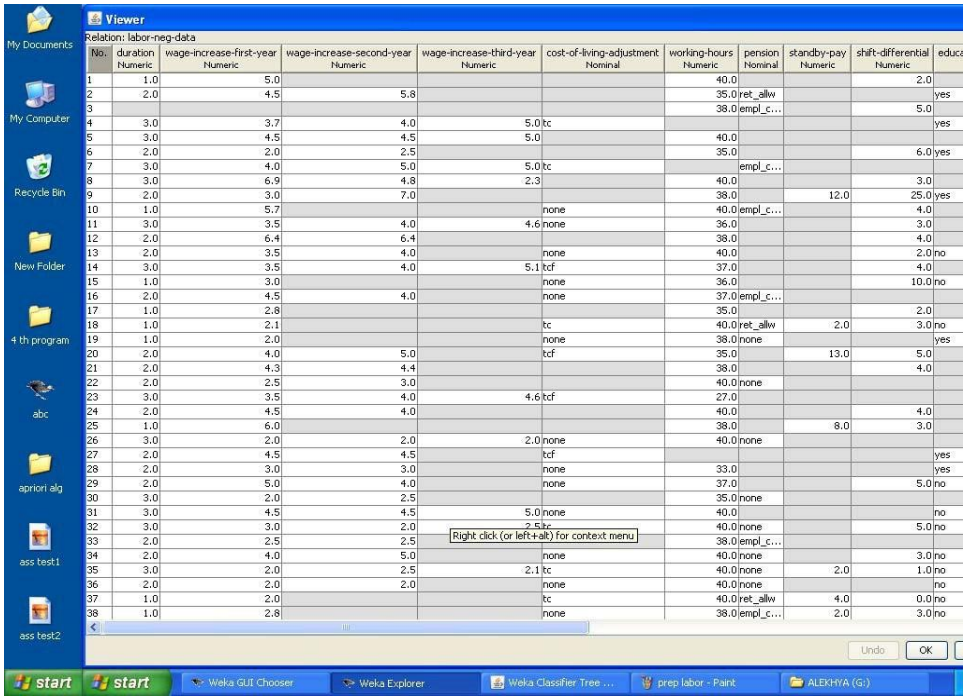
Left-Clicking on any of these gives an opportunity to alter the filters settings. For example, the setting may take a text string, in which case you type the string into the text field provided. Or it may give a drop-down box listing several states to choose from. Or it may do something else, depending on the information required. Information on the options is provided in a tool tip if you let the mouse pointer hover of the corresponding field. More information on the filter and its options can be obtained by clicking on the More button in the About panel at the top of the GenericObjectEditor window.

Applying Filters

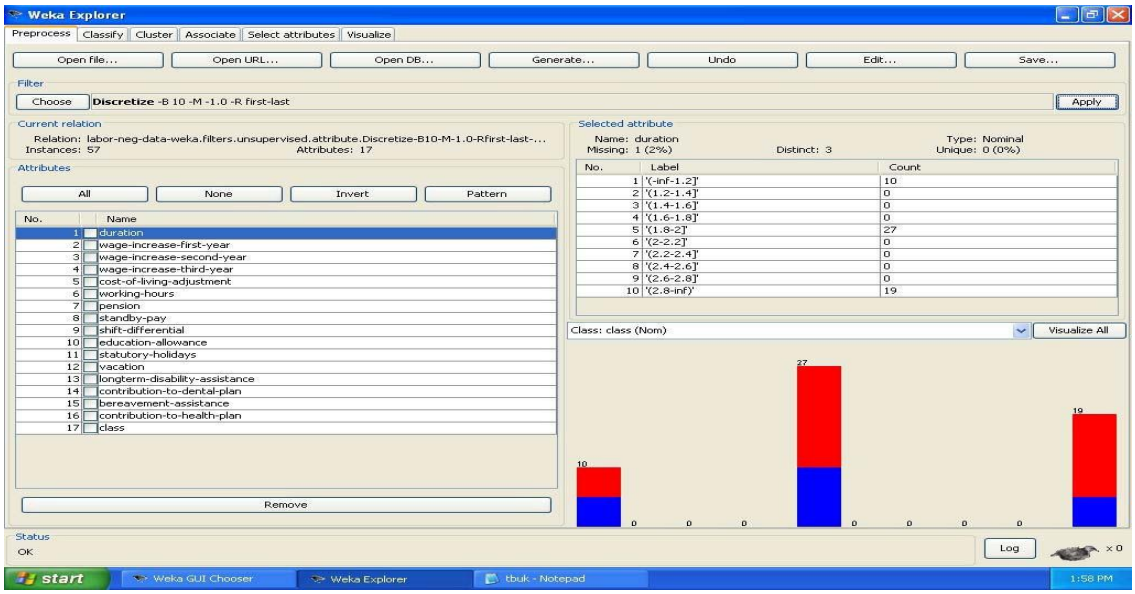
Once you have selected and configured a filter, you can apply it to the data by pressing the Apply button at the right end of the Filter panel in the Preprocess panel. The Preprocess panel will then show the transformed data. The change can be undone by pressing the Undo button. You can also use the Edit...button to modify your data manually in a dataset editor. Finally, the Save... button at the top right of the Preprocess panel saves the current version of the relation in file formats that can represent the relation, allowing it to be kept for future use.

➓ Steps for run preprocessing tab in WEKA

- Open WEKA Tool.
- Click on WEKA Explorer.
- Click on Preprocessing tab button.
- Click on open file button.
- Choose WEKA folder in C drive.
- Select and Click on data option button.
- Choose labor data set and open file.
- Choose filter button and select the Unsupervised-Discritize option and apply
- Dataset labor.arff



The following screenshot shows the effect of discretization



EXERCISE 4:

Explore various options in Weka for preprocessing data and apply in each dataset.

Eg: creditg,Soybean, Vote, Iris, Contactlense,

OUTPUT:

VIVA QUESTIONS:

1. List some applications of data mining.

Agriculture, biological data analysis, call record analysis, DSS, Business intelligence system etc

2. Why do we pre-process the data?

To ensure the data quality. [accuracy, completeness, consistency, timeliness, believability, interpret-ability]

3. What are the steps involved in data pre-processing?

Data cleaning, data integration, data reduction, data transformation.

4. Define virtual data warehouse.

A virtual data warehouse provides a compact view of the data inventory. It contains meta data and uses middle-ware to establish connection between different data sources.

5.Define KDD.

The process of finding useful information and patterns in data.

6.Define metadata.

A database that describes various aspects of data in the warehouse is called metadata.

7.What are data mining techniques?

a. Association rules
b. Classification and prediction
c. Clustering
d. Deviation detection
e. Similarity search

8.List the typical OLAP operations.

f. Roll UP
g. DRILL DOWN
h. ROTATE
i. SLICE AND DICE

B.Load each dataset into Weka and run Apriori algorithm with different support and confidence values. Study the rules generated.

AIM: To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best known constraints are minimum thresholds on support and confidence. The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the itemset. In the example database, the itemset {milk, bread} has a support of 2 / 5 = 0.4 since it occurs in 40% of all transactions (2 out of 5 transactions).

THEORY:

Association rule mining is defined as: Let be a set of n binary attributes called items. Let be a set of transactions called the database. Each transaction in D has a unique transaction ID and contains a subset of the items in I. A rule is defined as an implication of the form X=>Y where X,Y C I and X Π Y=Φ . The sets of items (for short itemsets) X and Y are called antecedent (left hand side or LHS) and consequent (right hand side or RHS) of the rule respectively.

To illustrate the concepts, we use a small example from the supermarket domain.

The set of items is I = {milk,bread,butter,beer} and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table to the right. An example rule for the supermarket could be meaning that if milk and bread is bought, customers also buy butter.

Note: this example is extremely small. In practical applications, a rule needs a support of several hundred transactions before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

To select interesting rules from the set of all possible rules, constraints on various measures of significance and interest can be used. The best known constraints are minimum thresholds on support and confidence. The support supp(X) of an itemset X is defined as the proportion of transactions in the data set which contain the

itemset. In the example database, the itemset {milk, bread} has a support of 2 / 5 = 0.4 since it occurs in 40% of all transactions (2 out of 5 transactions).
The confidence of a rule is defined. For example, the rule has a confidence of 0.2 / 0.4 = 0.5 in the database, which means that for 50% of the transactions containing milk and bread the rule is correct. Confidence can be interpreted as an estimate of the probability P(Y | X), the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

ALGORITHM:
Association rule mining is to find out association rules that satisfy the predefined minimum support and confidence from a given database. The problem is usually decomposed into two sub problems. One is to find those itemsets whose occurrences exceed a predefined threshold in the database; those itemsets are called frequent or large itemsets. The second problem is to generate association rules from those large itemsets with the constraints of minimal confidence.
Suppose one of the large itemsets is Lk, Lk = {I1, I2, … , Ik}, association rules with this itemsets are generated in the following way: the first rule is {I1, I2, … , Ik1} and {Ik}, by checking the confidence this rule can be determined as interesting or not.
Then other rule are generated by deleting the last items in the antecedent and inserting it to the consequent, further the confidences of the new rules are checked to determine the interestingness of them. Those processes iterated until the antecedent becomes empty.
Since the second subproblem is quite straight forward, most of the researches focus on the first subproblem.

The Apriori algorithm finds the frequent sets *L* In Database *D*.

· Find frequent set *Lk* − 1.
· Join Step.
. *Ck* is generated by joining *Lk* − 1with itself

· Prune Step.

o Any (*k* − 1) itemset that is not frequent cannot be a subset of a

frequent *k* itemset, hence should be removed.

Where · (*Ck*: Candidate itemset of size *k*)

· (*Lk*: frequent itemset of size *k*)

Apriori Pseudocode

Apriori (T,£)

L<{ Large 1itemsets that appear in more than transactions }
while L(k1)≠ Φ *C(k)<Generate*(*Lk* − 1) *for transactions* t € T

C(t)Subset(*Ck*,*t*)

for candidates c € C(t)

count[c]<count[c]+1 L(k)<{ c

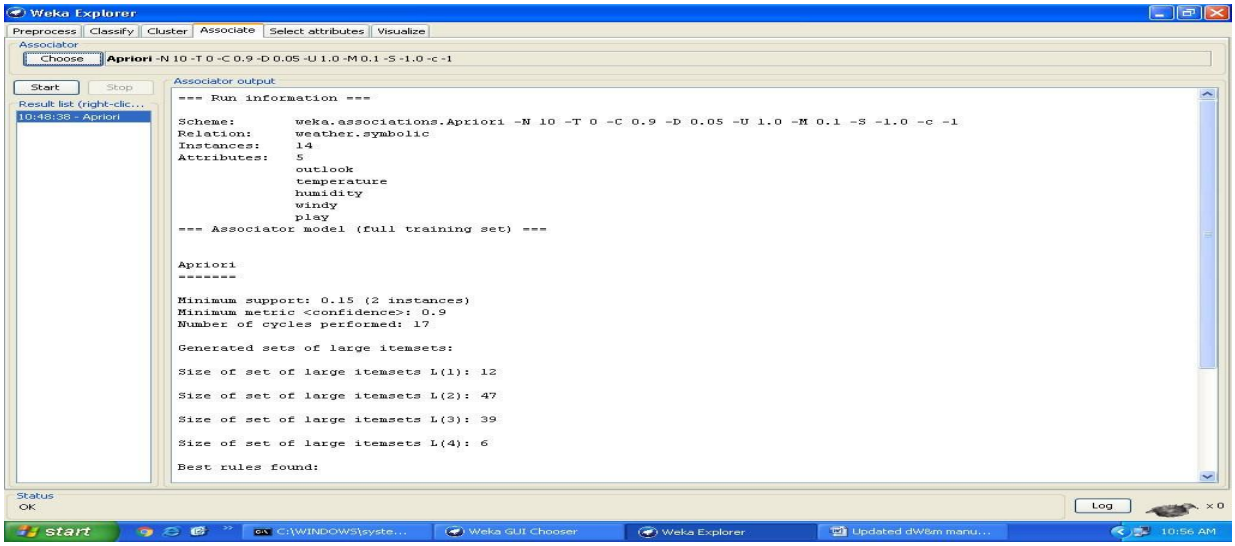
€ C(k)| count[c] ≥ £K<K+ 1

return Ụ L(k) k.

Steps for run Apriori algorithm in WEKA

- Open WEKA Tool.
- Click on WEKA Explorer.
  - Click on Preprocessing tab button.
  - Click on open file button.
  - Choose WEKA folder in C drive.
- Select and Click on data option button.
- Choose Weather data set and open file.
- Click on Associate tab and Choose Apriori algorithm
- Click on start button.

OUTPUT:



Association Rule:

An association rule has two parts, an antecedent (if) and a consequent (then). An antecedent is an item found in the data. A consequent is an item that is found in combination with the antecedent.

Association rules are created by analyzing data for frequent if/then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the database. Confidence indicates the number of times the if/then statements have been found to be true.

In data mining, association rules are useful for analyzing and predicting customer behavior. They play an important part in shopping basket data analysis, product clustering, catalog design and store layout.

Support and Confidence values:

- Support count: The support count of an itemset *X*, denoted by *X.count*, in a data set *T* is the number of transactions in *T* that contain *X*. Assume *T* has *n* transactions.
- Then,

$$support = \frac{(X \cup Y).count}{n}$$

$$confidence = \frac{(X \cup Y).count}{X.count}$$

support = support({*A* U *C*})

confidence = support({*A* U *C*})/support({*A*})

EXERCISE 5: Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated. Derive interesting insights and observe the effect of discretization in the rule generation process.

Eg:Dataset like Vote,soybean,supermarket,Iris..

Steps for run Apriori algorithm in WEKA

- Open WEKA Tool.
- Click on WEKA Explorer.
- Click on Preprocessing tab button.
- Click on open file button.
- Choose WEKA folder in C drive.
- Select and Click on data option button.
- Choose Weather data set and open file.
- Choose filter button and select the Unsupervised-Discritize option and apply
- Click on Associate tab and Choose Aprior algorithm
- Click on start button.

Viva voice questions

1. What is the difference between dependent data warehouse and independent data warehouse?

There is a third type of Datamart called Hybrid. The Hybrid datamart having source data from Operational systems or external files and central Datawarehouse as well. I will definitely check for Dependent and Independent Datawarehouses and update.

2. Explain Association algorithm in Data mining?

Association algorithm is used for recommendation engine that is based on a market based analysis. This engine suggests products to customers based on what they bought earlier. The model is built on a dataset containing identifiers. These identifiers are both for individual cases and for the items that cases contain. These groups of items in a data set are called as an item set. The algorithm traverses a data set to find items that appear in a case. MINIMUM_SUPPORT parameter is used any associated items that appear into an item set.

3. What are the goals of data mining?

Prediction, identification, classification and optimization

4. What are data mining functionality?

   Mining frequent pattern, association rules, classification and prediction, clustering, evolution analysis and outlier Analysis

5. If there are 3 dimensions, how many cuboids are there in cube?
   2^3 = 8 cuboids

6. Define support and confidence.

   The support for a rule R is the ratio of the number of occurrences of R, given all occurrences of all rules.The confidence of a rule X->Y, is the ratio of the number of occurrences of Y given X, among all other occurrences given X.

7. What is the main goal of data mining?

   The main goal of data mining is Prediction.

SIGNATURE OF FACULTY

WEEK– 3 : Demonstrate performing classification on data sets.

AIM: Implementing the decision tree analysis and the training data in the data set.

THEORY:

Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data. For example, a classification model could be used to identify loan applicants as low, medium, or high credit risks. A classification task begins with a data set in which the class assignments are known. For example, a classification model that predicts credit risk could be developed based on observed data for many loan applicants over a period of time.

In addition to the historical credit rating, the data might track employment history, home ownership or rental, years of residence, number and type of investments, and so on. Credit rating would be the target, the other attributes would be the predictors, and the data for each customer would constitute a case.

Classifications are discrete and do not imply order. Continuous, floating point values would indicate a numerical, rather than a categorical, target. A predictive model with a numerical target uses a regression algorithm, not a classification algorithm. The simplest type of classification problem is binary classification. In binary classification, the target attribute has only two possible values: for example, high credit rating or low credit rating. Multiclass targets have more than two values: for example, low, medium, high, or unknown credit rating. In the model build (training) process, a classification algorithm finds relationships between the values of the predictors and the values of the target. Different classification algorithms use different techniques for finding relationships. These relationships are summarized in a model, which can then be applied to a different data set in which the class assignments are unknown

Different Classification Algorithms: Oracle Data Mining provides the following algorithms for classification:

- Decision Tree - Decision trees automatically generate rules, which are conditional statements that reveal the logic used to build the tree.

- Naive Bayes - Naive Bayes uses Bayes' Theorem, a formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

Classification Tab

Selecting a Classifier

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

Test Options

The result of applying the chosen classifier will be tested according to the options that are set by clicking in the Test options box. There are four test modes:

1. Use training set. The classifier is evaluated on how well it predicts the class of the instances it was trained on.

2. Supplied test set. The classifier is evaluated on how well it predicts the class of a set of instances loaded from a file. Clicking the Set... button brings up a dialog allowing you to choose the file to test on.

3. Cross-validation. The classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field.

4. Percentage split. The classifier is evaluated 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.

Classifier Evaluation Options:

1. Output model. The classification model on the full training set is output so that it can be viewed, visualized, etc. This option is selected by default.

2. Output per-class stats. The precision/recall and true/false statistics for each class are output. This option is also selected by default.

3. Output entropy evaluation measures. Entropy evaluation measures are included in the output. This option is not selected by default.

4. Output confusion matrix. The confusion matrix of the classifier's predictions is included in the output. This option is selected by default.

5. Store predictions for visualization. The **classifier's predictions are** remembered so that they can be visualized. This option is selected by default.

6. Output predictions. The predictions on the evaluation data are output.

Note that in the case of a cross-validation the instance numbers do not correspond to the location in the data!

7. Output additional attributes. If additional attributes need to be output alongside the

predictions, e.g., an ID attribute for tracking misclassifications, then the index of this attribute can be specified here. The usual **Weka ranges are supported,"first" and "last" are therefore valid** indices as **well (example: "first**-3,6,8,12-**last").**

8. Cost-sensitive evaluation. The errors is evaluated with respect to a cost matrix. The Set... button allows you to specify the cost matrix used.

9. Random seed for xval / % Split. This specifies the random seed used when randomizing the data before it is divided up for evaluation purposes.

10. Preserve order for % Split. This suppresses the randomization of the data before splitting into train and test set.

11. Output source code. If the classifier can output the built model as Java source code, you can specify the class name here. The code will be printed **in the "Classifier output" area.**

The Class Attribute

The classifiers in **WEKA are designed to be trained to predict a single 'class'** attribute, which is the target for prediction. Some classifiers can only learn nominal classes; others can only learn numeric classes (regression problems) still others can learn both.

By default, the class is taken to be the last attribute in the data. If you want to train a classifier to predict a different attribute, click on the box below the Test options box to bring up a drop-down list of attributes to choose from.

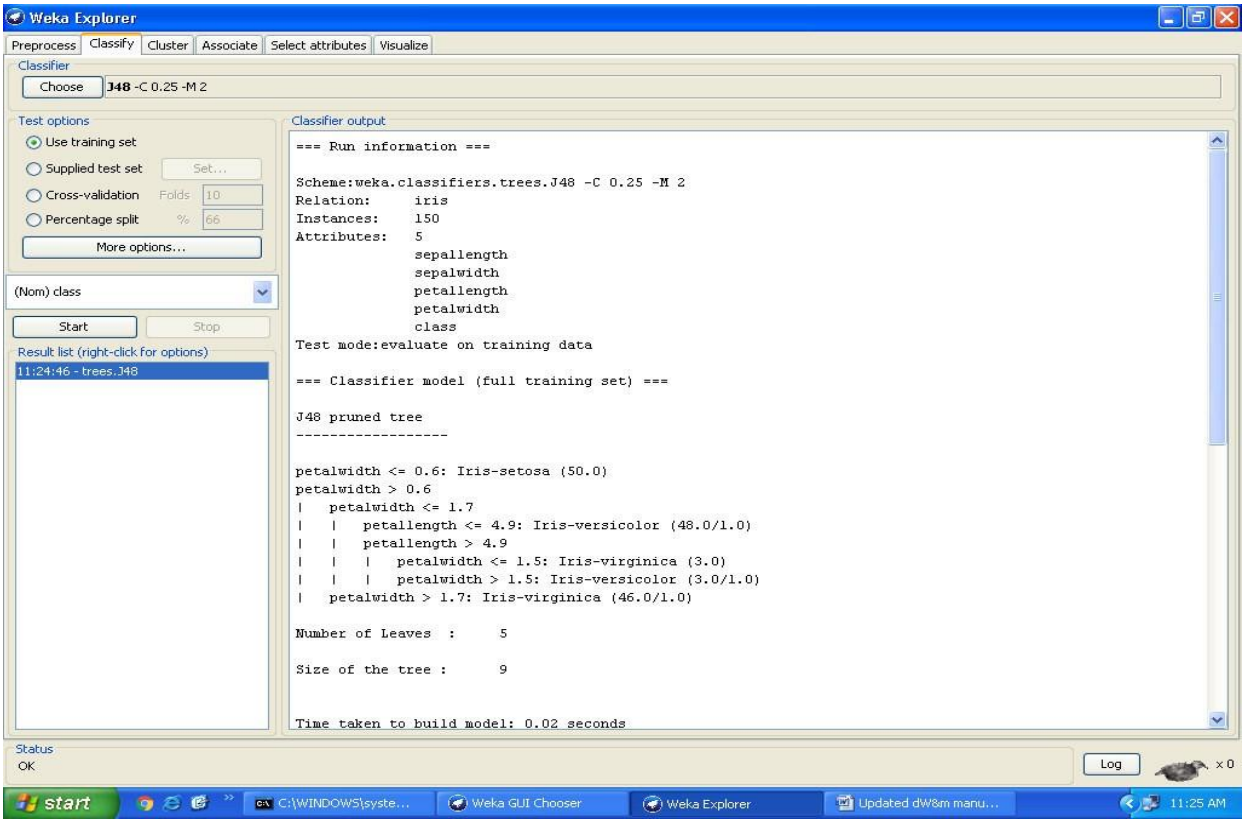
Training a Classifier

Once the classifier, test options and class have all been set, the learning process is started by clicking on the Start button. While the classifier is busy being trained, the little bird moves around. You can stop the training process at any time by clicking on the Stop button. When training is complete, several things happen. The Classifier output area to the right of the display is filled with text describing the results of training and testing. A new entry appears in the Result list box. We look at the result list below; but first we investigate the text that has been output.

A.Load each dataset into Weka and run id3, j48 classification algorithm, study the classifier output. Compute entropy values, Kappa statistic.

Ans:

- Steps for run ID3 and J48 Classification algorithms in WEKA
  - Open WEKA Tool.
  - Click on WEKA Explorer.
    - Click on Preprocessing tab button.
    - Click on open file button.
    - Choose WEKA folder in C drive.
  - Select and Click on data option button.
  - Choose iris data set and open file.
  - Click on classify tab and Choose J48 algorithm and select use training set test option.
  - Click on start button.
  - Click on classify tab and Choose ID3 algorithm and select use training set test option.
  - Click on start button.



The Classifier Output Text

The text in the Classifier output area has scroll bars allowing you to browse the results. Clicking with the left mouse button into the text area, while holding Alt and Shift, brings up a dialog that enables you to save the displayed output

in a variety of formats (currently, BMP, EPS, JPEG and PNG). Of course, you can also resize the Explorer window to get a larger display area.

The output is

Split into several sections:

1. Run information. A list of information giving the learning scheme options, relation name, instances, attributes and test mode that were involved in the process.

2. Classifier model (full training set). A textual representation of the classification model that was produced on the full training data.

3. The results of the chosen test mode are broken down thus.

4. Summary. A list of statistics summarizing how accurately the classifier was able to predict the true class of the instances under the chosen test mode.

5. Detailed Accuracy By Class. A more detailed per-class break down **of the classifier's** prediction accuracy.

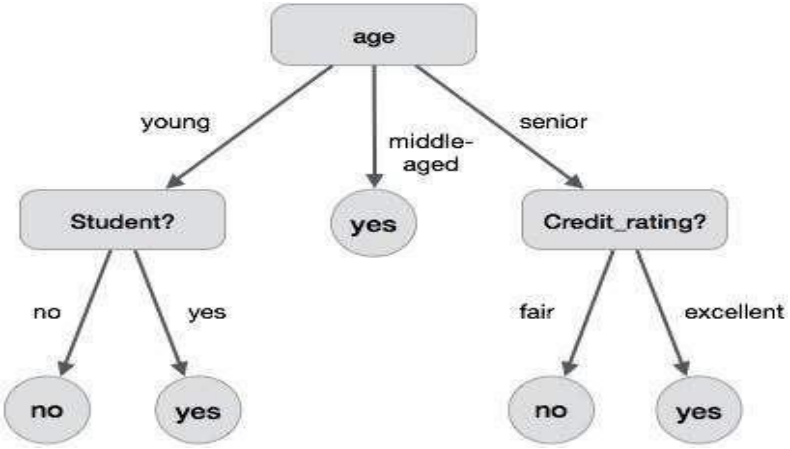
6. Confusion Matrix. Shows how many instances have been assigned to each class. Elements show the number of test examples whose actual class is the row and whose predicted class is the column.

7. Source code (optional). This section lists the Java source code if one chose "Output source code" in the "More options" dialog.

B. extract if-then rues from decision tree generated by classifier, Observe the confusion matrix and derive Accuracy, F- measure, TPrate, FPrate , Precision and recall values. Apply cross-validation strategy with various fold levels and compare the accuracy results.

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy _computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

IF-THEN Rules:

Rule-based classifier makes use of a set of IF-THEN rules for classification. We can express a rule in the following from −

IF condition THEN conclusion Let us consider a rule R1,

R1: IF age=youth AND student=yes

THEN buy_computer=yes

Points to remember −

- The IF part of the rule is called **rule antecedent** or **precondition**.
- The THEN part of the rule is called **rule consequent**.
- The antecedent part the condition consist of one or more attribute tests and these tests are logically ANDed.

- The consequent part consists of class prediction.

− We can also write rule R1 as follows:

```
R1: (age = youth) ^ (student = yes))(buys computer = yes)
```

If the condition holds true for a given tuple, then the antecedent is satisfied.

Rule Extraction

Here we will learn how to build a rule-based classifier by extracting IF-THEN rules from a decision tree.

Points to remember −

- One rule is created for each path from the root to the leaf node.
- To form a rule antecedent, each splitting criterion is logically ANDed.
- The leaf node holds the class prediction, forming the rule consequent.

Rule Induction Using Sequential Covering Algorithm

Sequential Covering Algorithm can be used to extract IF-THEN rules form the training data. We do not require to generate a decision tree first. In this algorithm, each rule for a given class covers many of the tuples of that class.

Some of the sequential Covering Algorithms are AQ, CN2, and RIPPER. As per the general strategy the rules are learned one at a time. For each time rules are learned, a tuple covered by the rule is removed and the process continues for the rest of the tuples. This is because the path to each leaf in a decision tree corresponds to a rule.

Note − The Decision tree induction can be considered as learning a set of rules simultaneously.

The Following is the sequential learning Algorithm where rules are learned for one class at a time. When learning a rule from a class Ci, we want the rule to cover all the tuples from class C only and no tuple form any other class.

Algorithm: Sequential Covering

Input:
D, a data set class-labeled tuples,

Att_vals, the set of all attributes and their possible values.
Output: A Set of IF-THEN rules. Method:
Rule_set={ }; // initial set of rules learned is empty for each

class c do

repeat
Rule = Learn_One_Rule(D, Att_valls, c); remove
tuples covered by Rule form D; until termination
condition;

Rule_set=Rule_set+Rule; // add a new rule to rule-set end for
return Rule_Set;

Rule Pruning

The rule is pruned is due to the following reason −

- The Assessment of quality is made on the original set of training data. The rule may perform well on training data but less well on subsequent data. That's why the rule pruning is required.
- The rule is pruned by removing conjunct. The rule R is pruned, if pruned version of R has greater quality than what was assessed on an independent set of tuples.

FOIL is one of the simple and effective method for rule pruning. For a given rule R,

FOIL_Prune = pos - neg / pos + neg
where pos and neg is the number of positive tuples covered by R, respectively.

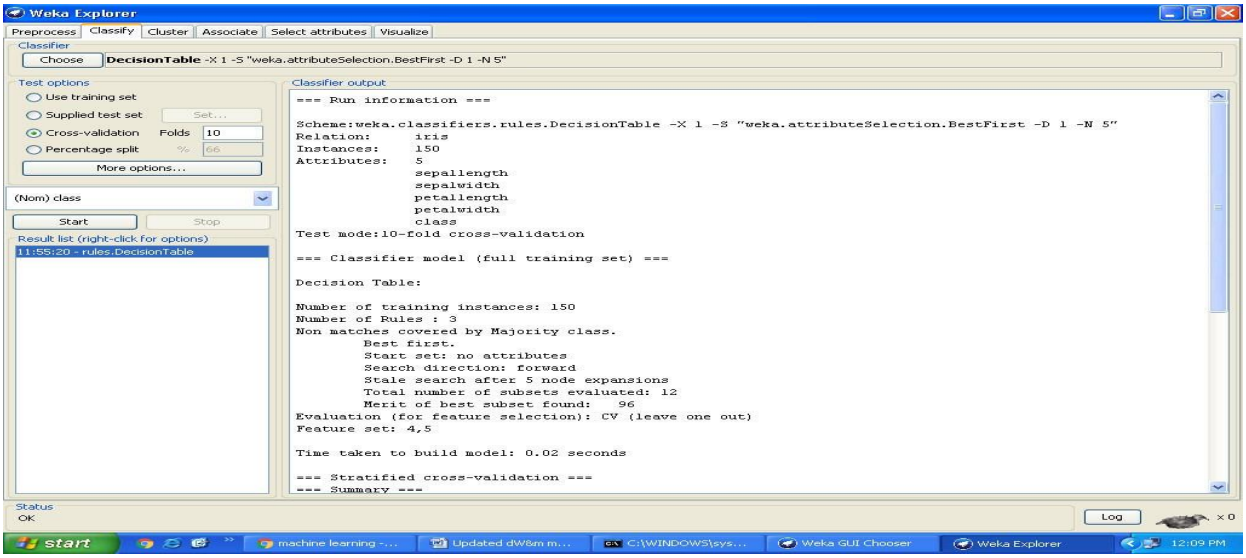
Note − This value will increase with the accuracy of R on the pruning set. Hence, if the FOIL_Prune value is higher for the pruned version of R, then we prune R.

➊ Steps for run decision tree algorithms in WEKA

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on Preprocessing tab button.
1. Click on open file button.
2. Choose WEKA folder in C drive.

3. Select and Click on data option button.
4. Choose iris data set and open file.
5. Click on classify tab and Choose decision table algorithm and select cross-validation folds value-10 test option.
6. Click on start button.

OUTPUT:



EXERCISE 6: Load each dataset into Weka and run id3, j48 classification algorithm, study the classifier output with available Datasets.

OUTPUT:

Viva voice questions

1. What is a Decision Tree Algorithm?

A decision tree is a tree in which every node is either a leaf node or a decision node. This tree takes an input an object and outputs some decision. All Paths from root node to the leaf node are reached by either using AND or OR or BOTH. The tree is constructed using the regularities of the data. The decision tree is not affected by Automatic Data Preparation.

2. What are issues in data mining?

Issues in mining methodology, performance issues, user interactive issues, different source of data types issues etc.

3. List some applications of data mining.

Agriculture, biological data analysis, call record analysis, DSS, Business intelligence system etc.

SIGNATURE OF FACULTY:

C. Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbor classification, Interpret the results obtained.

AIM: Determining and classifying the credit good or bad in the dataset with an Accuracy.

THEORY:

Naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even though these features depend on the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

An advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the parameters (means and variances of the variables) necessary for classification. Because independent variables are assumed, only the variances of the variables for each class need to be determined and not the entire covariance matrix The naive Bayes probabilistic model :

The probability model for a classifier is a conditional model.

P(C|F1 .....Fn) over a dependent class variable *C* with a small number of outcomes or *classes*, conditional on several feature variables *F*1 through *Fn*. The problem is that if the number of features *n* is large or when a feature can take on a large number of values, then basing such a model on probability tables is infeasible. We therefore reformulate the model to make it more tractable.

Using Bayes‟ theorem, we write

P(C|F1.....Fn)=[{p(C)p(F1.....Fn|C)}/p(F1,..... Fn)]

In plain English the above equation can be written as

Posterior= [(prior *likehood)/evidence]

In practice we are only interested in the numerator of that fraction, since the denominator does not depend on *C* and the values of the features *Fi* are given, so that the denominator is effectively constant.

Now the "naive" conditional independence assumptions come into play: assume that each feature *Fi* is conditionally independent of every other feature *Fj* .

This means that p(Fi|C,Fj)=p(Fi|C) and so the joint model can be expressed as p(C,F1,.....Fn)=p(C)p(F1|C)p(F2|C)..... = p(C)π p(Fi|C).

This means that under the above independence assumptions, the conditional distribution over the class variable *C* can be expressed like this:

p(C|F1. .....Fn)= p(C) πp(Fi|C) Z

where *Z* is a scaling factor dependent only on F1. ..... Fn, i.e., a constant if the values of the feature variables are known.

Models of this form are much more manageable, since they factor into a so called *class prior p*(*C*) and independent probability distributions p(Fi|C). If there are *k* classes and if a model for each p(Fi|C=c) can be expressed in terms of *r* parameters, then the corresponding naive Bayes model has (*k* − 1) + *n r k* parameters. In practice, often *k* = 2 (binary classification) and *r* = 1 (Bernoulli variables as features) are common, and so the total number of parameters of the naive Bayes model is 2*n* + 1, where *n* is the number of binary features used for prediction

P(h/D)= P(D/h) P(h) P(D)

- P(h) : Prior probability of hypothesis h
- P(D) : Prior probability of training data D
- P(h/D) : Probability of h given D
- P(D/h) : Probability of D given h

Naïve Bayes Classifier : Derivation

• D : Set of tuples

– Each Tuple is an „n" dimensional attribute vector

– X : (x1,x2,x3,…. xn)

- Let there me „m" Classes : C1,C2,C3…Cm
- NB classifier predicts X belongs to Class Ci iff

– P (Ci/X) > P(Cj/X) for 1<= j <= m , j <> i

- Maximum Posteriori Hypothesis

– P(Ci/X) = P(X/Ci) P(Ci) / P(X)

– Maximize P(X/Ci) P(Ci) as P(X) is

- With many attributes, it is computationally expensive to evaluate P(X/Ci)
- Naïve Assumption of "class conditional independence"
- P(X/Ci) = n P(xk/ Ci)

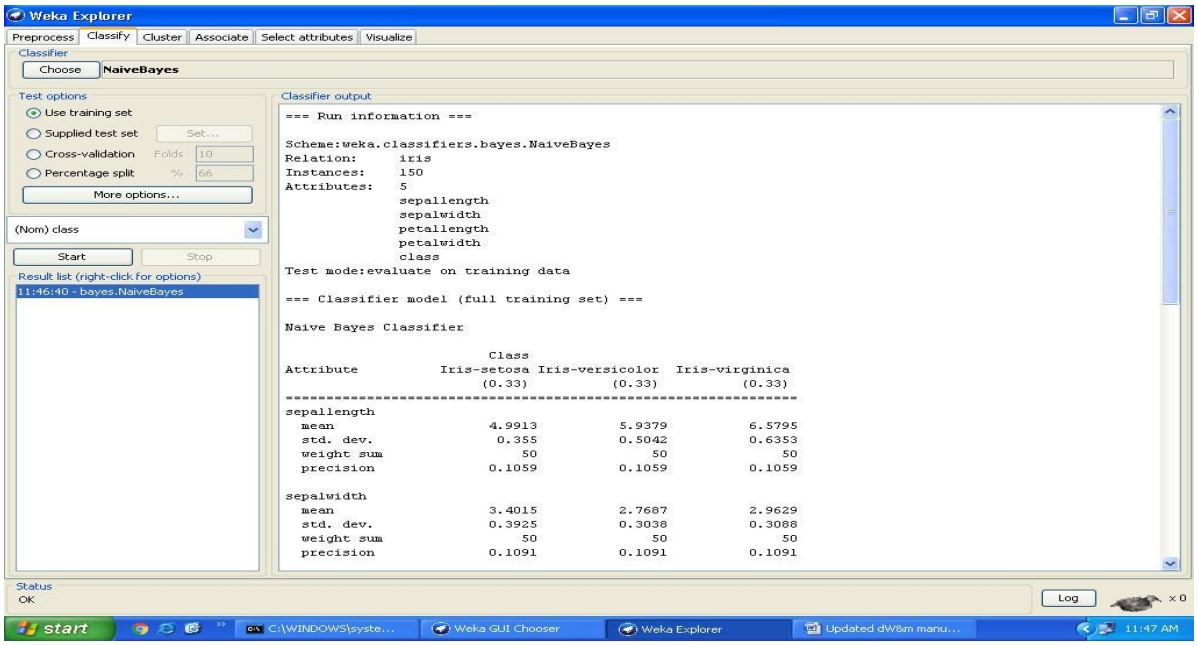
• P(X/Ci) = P(x1/Ci) * P(x2/Ci) *…* P(xn/ Ci).

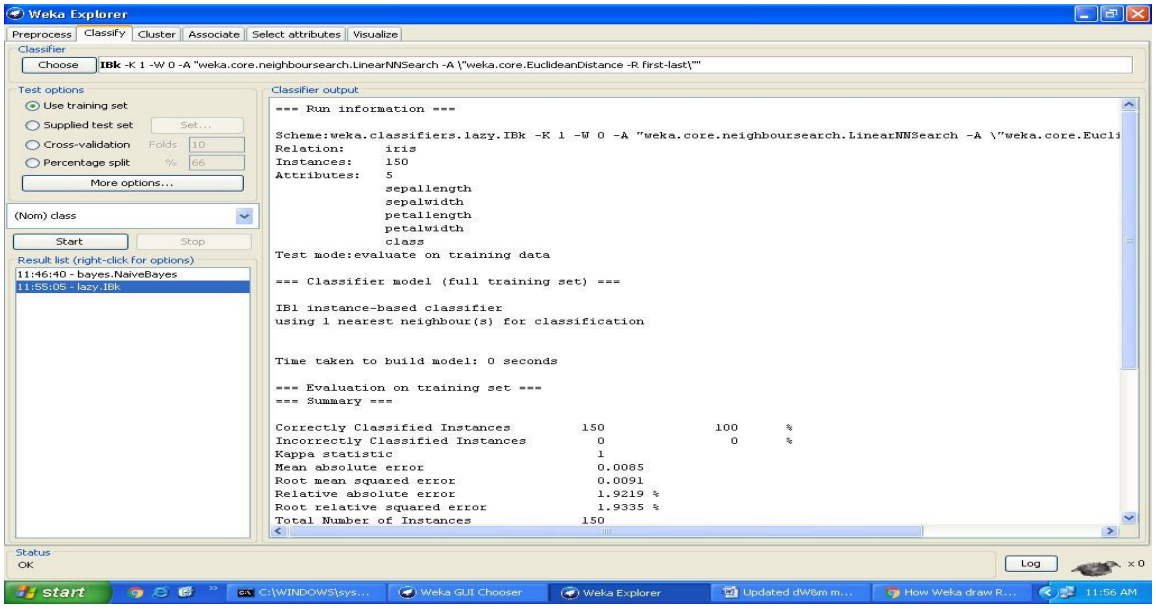
➊ Steps for run Naïve-bayes and k-nearest neighbor Classification algorithms in WEKA

- o Open WEKA Tool.
- o Click on WEKA Explorer.
  - Click on Preprocessing tab button.
  - Click on open file button.
  - Choose WEKA folder in C drive.
- o Select and Click on data option button.
- o Choose iris data set and open file.
- o Click on classify tab and Choose Naïve-bayes algorithm and select use training set test option.

- Click on start button.
- Click on classify tab and Choose k-nearest neighbor and select use training set test option.
- Click on start button.

OUTPUT:

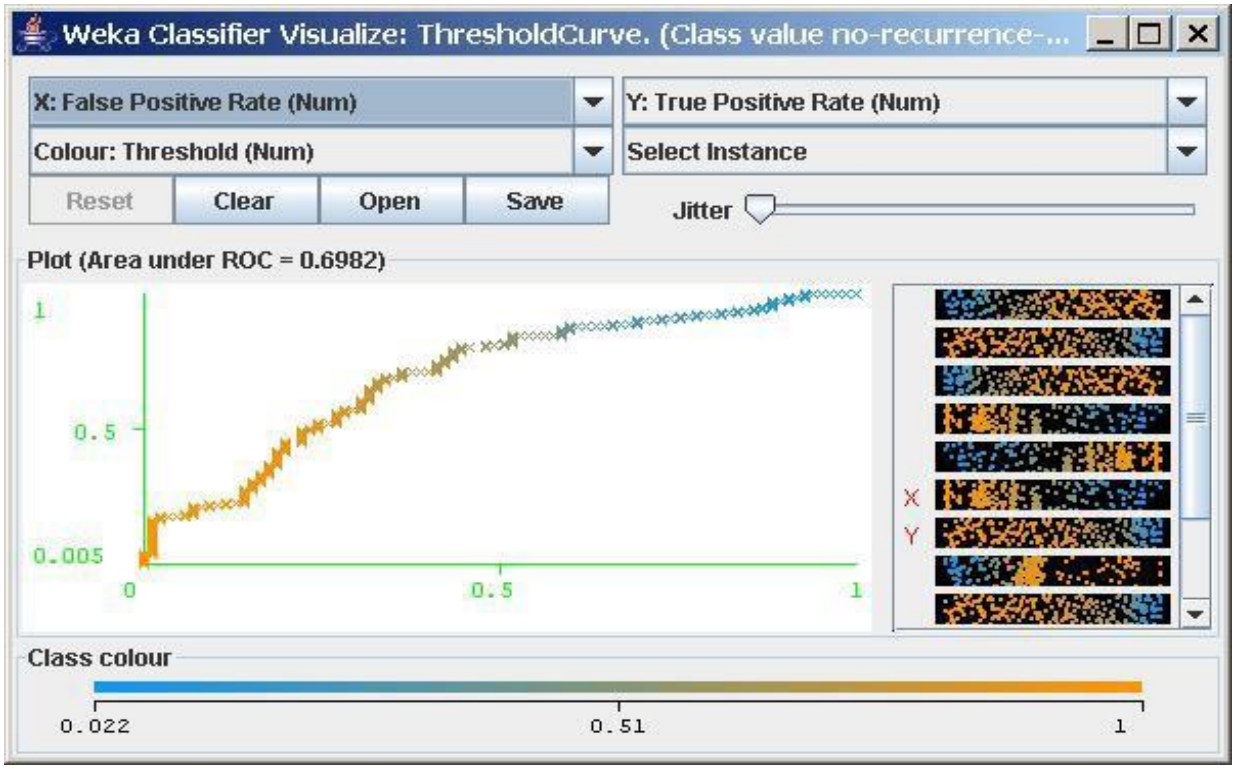




Plot RoC Curves.

Ans: Steps for identify the plot RoC Curves.

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on Visualize button.
4. Click on right click button.
5. Select and Click on polyline option button.



EXERCISE:7

Compare classification results of ID3,J48, Naïve-Bayes and k-NN classifiers for each dataset , and reduce which classifier is performing best and poor for each dataset and justify.

- Steps for run ID3 and J48 Classification algorithms in WEKA

  - Open WEKA Tool.
  - Click on WEKA Explorer.
  - Click on Preprocessing tab button.
  - Click on open file button.
  - Choose WEKA folder in C drive.

- Select and Click on data option button.
- Choose iris data set and open file.
- Click on classify tab and Choose J48 algorithm and select use training set test option.
- Click on start button.
- Click on classify tab and Choose ID3 algorithm and select use training set test option.
- Click on start button.
- Click on classify tab and Choose Naïve-bayes algorithm and select use training set test option.
- Click on start button.
- Click on classify tab and Choose k-nearest neighbor and select use training set test option.
- Click on start button.

OUTPUT:

Viva voice questions

1. What is K-nearest neighbor algorithm?
   It is one of the lazy learner algorithm used in classification. It finds the k-nearest neighbor of the point of interest.

2. **What are the issues regarding classification and prediction?**

   Preparing data for classification and prediction
   Comparing classification and prediction

3. What is decision tree classifier?

A decision tree is an hierarchically based classifier which compares data with a range of properly selected features.

4. What is multimedia data mining?

Multimedia Data Mining is a subfield of data mining that deals with an extraction of implicit knowledge, multimedia data relationships, or other patterns not explicitly stored in multimedia databases.

5. What is text mining?

Text mining is the procedure of synthesizing information, by analyzing relations, patterns, and rules among textual data. These procedures contains text summarization, text categorization, and text clustering.

6. What is Naïve Bayes Algorithm?

Naïve Bayes Algorithm is used to generate mining models. These models help to identify relationships between input columns and the predictable columns. This algorithm can be used in the initial stage of exploration. The algorithm calculates the probability of every state of each input column given predictable columns possible states. After the model is made, the results can be used for exploration and making predictions.

7. What is distributed data warehouse?

Distributed data warehouse shares data across multiple data repositories for the purpose of OLAP operation.

8. What is are different data warehouse model?

Enterprise data ware house
Data marts
Virtual Data warehouse

9. What are issues in data mining?

Issues in mining methodology, performance issues, user interactive issues, different source of data types issues etc

10. What are frequent pattern?
    a. A set of items that appear frequently together in a transaction data set.
    b. eg milk, bread, sugar

SIGNATURE OF FACULTY:

WEEK – 4: demonstrate performing clustering on data sets Clustering Tab

AIM: To understanding the selected attributes and removing attributes also to reload & the arff data file to get all the attributes in the data set.

Selecting a Clusterer

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

Cluster Modes

The Cluster mode box is used to choose what to cluster and how to evaluate

the results. The first three options are the same as for classification: Use training set, Supplied test set and Percentage split (Section 5.3.1)—except that now the data is assigned to clusters instead of trying to predict a specific class. The fourth mode, Classes to clusters evaluation, compares how well the chosen clusters match up with a pre-assigned class in the data. The drop-down box below this option selects the class, just as in the Classify panel.

An additional option in the Cluster mode box, the Store clusters for visualization tick box, determines whether or not it will be possible to visualize the clusters once training is complete. When dealing with datasets that are so large that memory becomes a problem it may be helpful to disable this option.

Ignoring Attributes

Often, some attributes in the data should be ignored when clustering. The Ignore attributes button brings up a small window that allows you to select which attributes are ignored. Clicking on an attribute in the window highlights it, holding down the SHIFT key selects a range

of consecutive attributes, and holding down CTRL toggles individual attributes on and off. To cancel the selection, back out with the Cancel button. To activate it, click the Select button. The next time clustering

is invoked, the selected attributes are ignored.

Working with Filters

The Filtered Clusterer meta-clusterer offers the user the possibility to apply filters directly before the clusterer is learned. This approach eliminates the manual application of a filter in the Preprocess panel, since the data gets processed on the fly. Useful if one needs to try out different filter setups.

Learning Clusters

The Cluster section, like the Classify section, has Start/Stop buttons, a result text area and a result list. These all behave just like their classification counterparts. Right-clicking an entry in the result list brings up a similar menu, except that it shows only two visualization options: Visualize cluster assignments and Visualize tree. The latter is grayed out when it is not applicable.

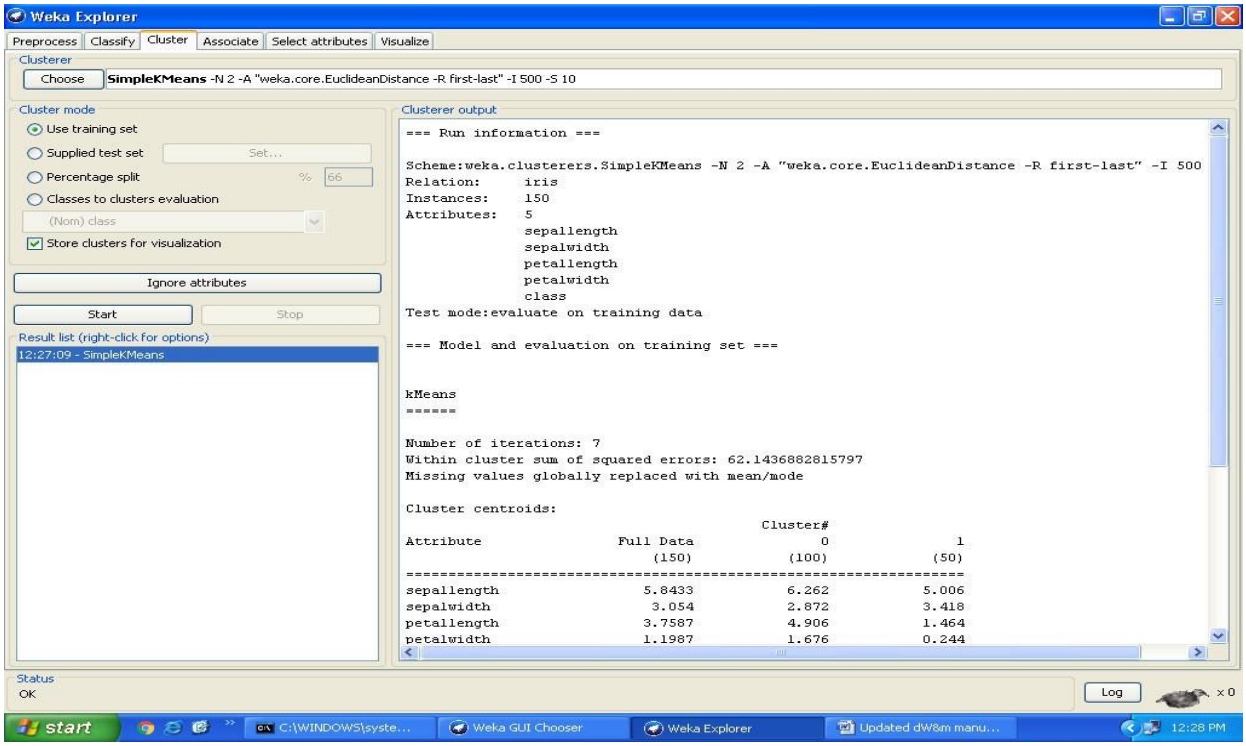
A.Load each dataset into Weka and run simple k-means clustering algorithm with different values of k(number of desired clusters). Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.

Ans:

- Steps for run K-mean Clustering algorithms in WEKA

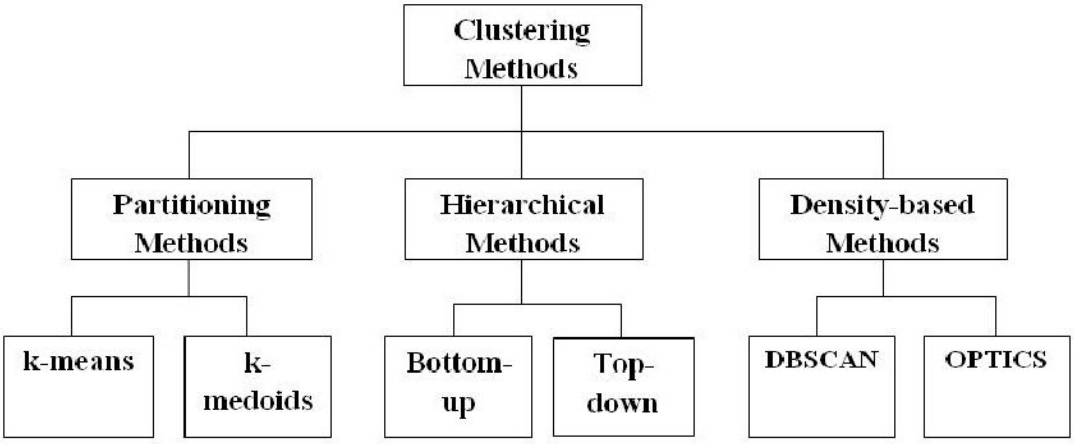
  - Open WEKA Tool.
  - Click on WEKA Explorer.
  - Click on Preprocessing tab button.
  - Click on open file button.
  - Choose WEKA folder in C drive.
  - Select and Click on data option button.
  - Choose iris data set and open file.
  - Click on cluster tab and Choose k-mean and select use training set test option.
  - Click on start button.

OUTPUT:



B.**Explore other clustering techniques available in Weka.**

AIM: Clustering Algorithms And Techniques in WEKA, They are



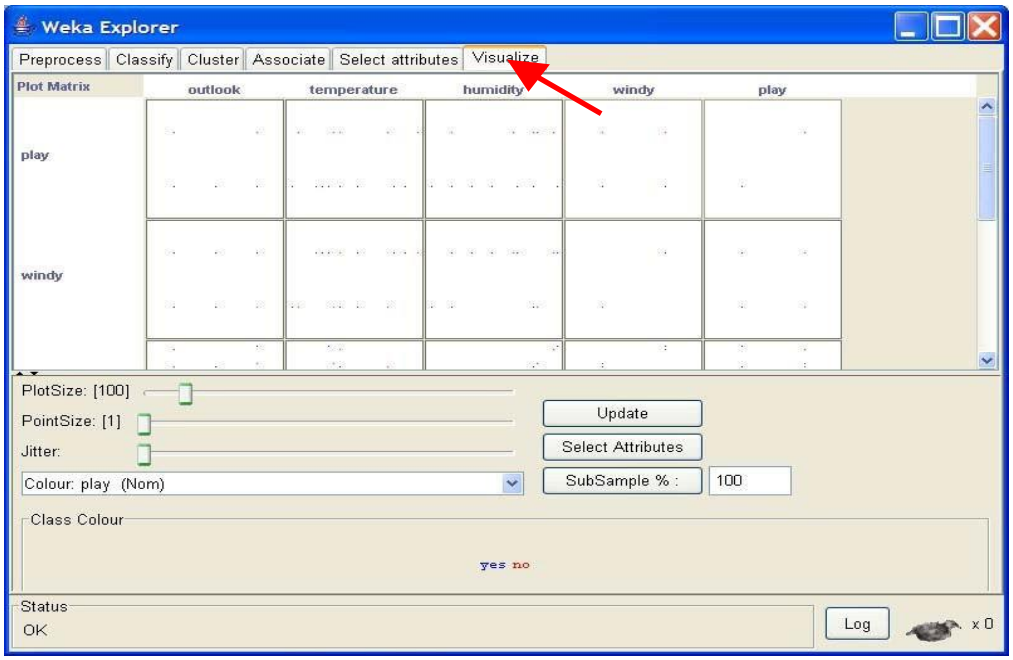
B. **Explore visualization features of weka to visualize the clusters. Derive interesting insights and explain.**

Visualize Features

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.

Access To **Visualization** From The *Classifier*, *Cluster* And *Attribute Selection* Panel Is Available From A Popup Menu. Click The Right Mouse Button Over An Entry In The Result List To Bring Up The Menu. You Will Be Presented With Options For Viewing Or Saving The Text Output And
--- Depending On The Scheme --- Further Options For Visualizing Errors, Clusters, Trees Etc.

To open Visualization screen, click 'Visualize' tab.

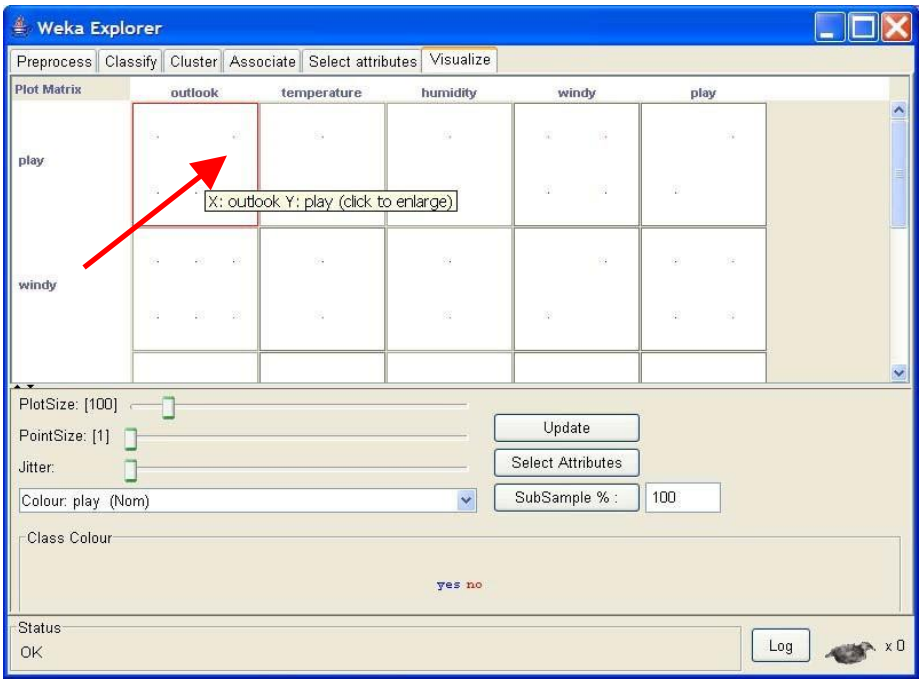


Select a square that corresponds to the attributes you would like to visualize. For example, let's choose 'outlook' for X – axis and 'play' for Y – axis. Click anywhere inside the square that corresponds to 'play o

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 colorpalette.n the left and 'outlook' at the top.



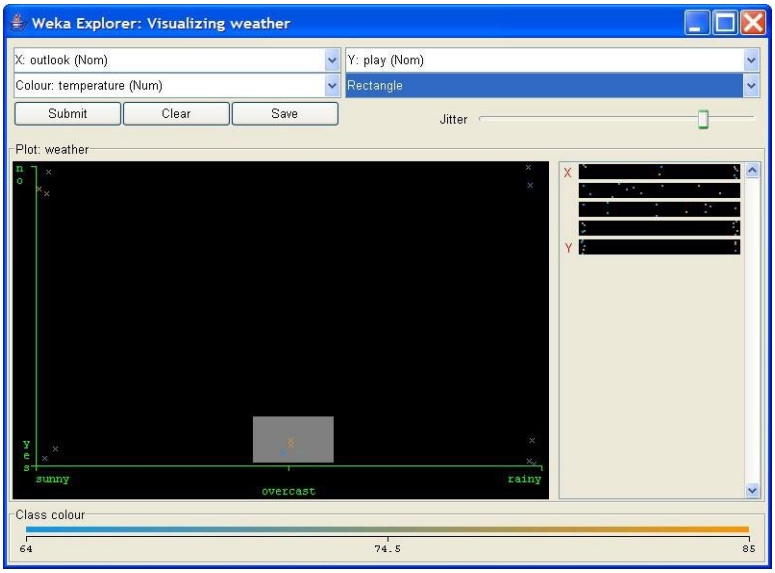
Selecting Instances

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]:

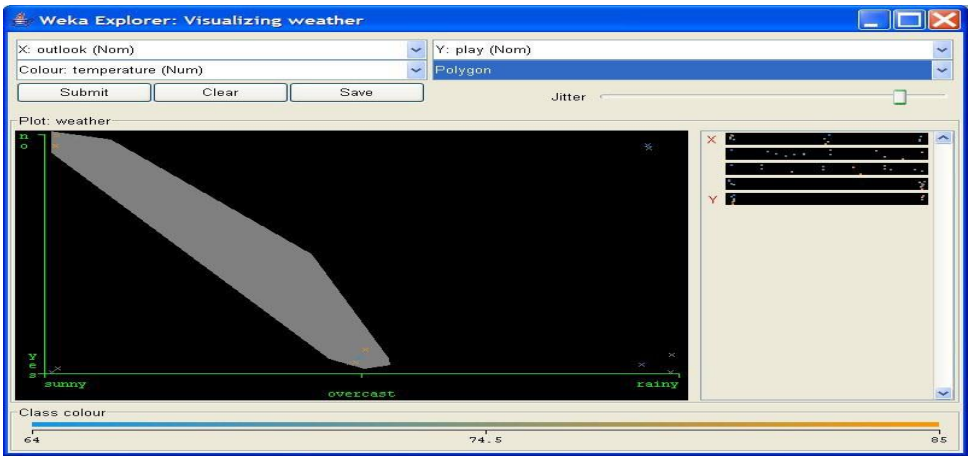
1. **Select Instance**. Click on an individual data point. It brings up a window listing

   attributes of the point. If more than one point will appear at the same location, more than one set of attributes will be shown.

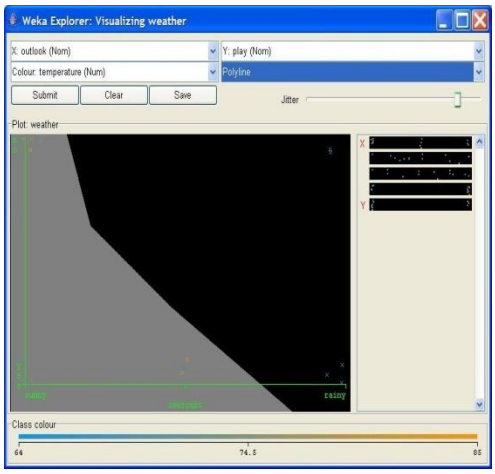
2. **Rectangle**. You can create a rectangle by dragging it around the point.

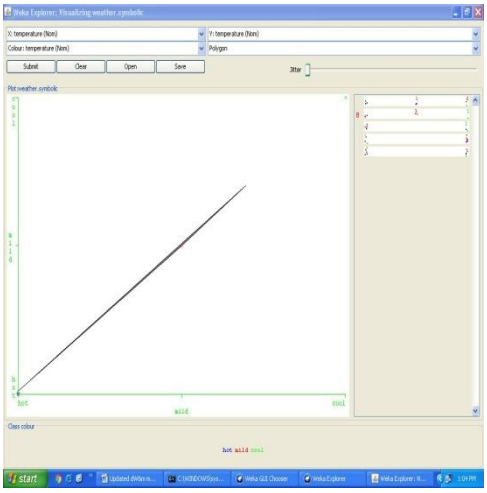


3. **Polygon**. You can select several points by building a free-form polygon. Left-click on the graph to add vertices to the polygon and right-click to complete it.



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.





SIGNATURE OF FACULTY:

WEEK-5: Sample Programs using German Credit Data.

Task 1: Credit Risk Assessment

Description: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide **whether the credit of a customer is good. Or bad. A bank's business rules regarding loans must** consider two opposing factors. On th one han, a bank wants to make as many loans as possible.

Interest on these loans is the banks profit source. On the other hand, a bank can not afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. Th**e bank's** loan policy must involved a compromise. Not too strict and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.

1. Knowledge engineering: Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in a number of ways.

2. Books: Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text from to production rule form.

3. Common sense: Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.

4. Case histories: Find records of actual cases where competent loan officers correctly judged when and not to. Approve a loan application.

The German Credit Data

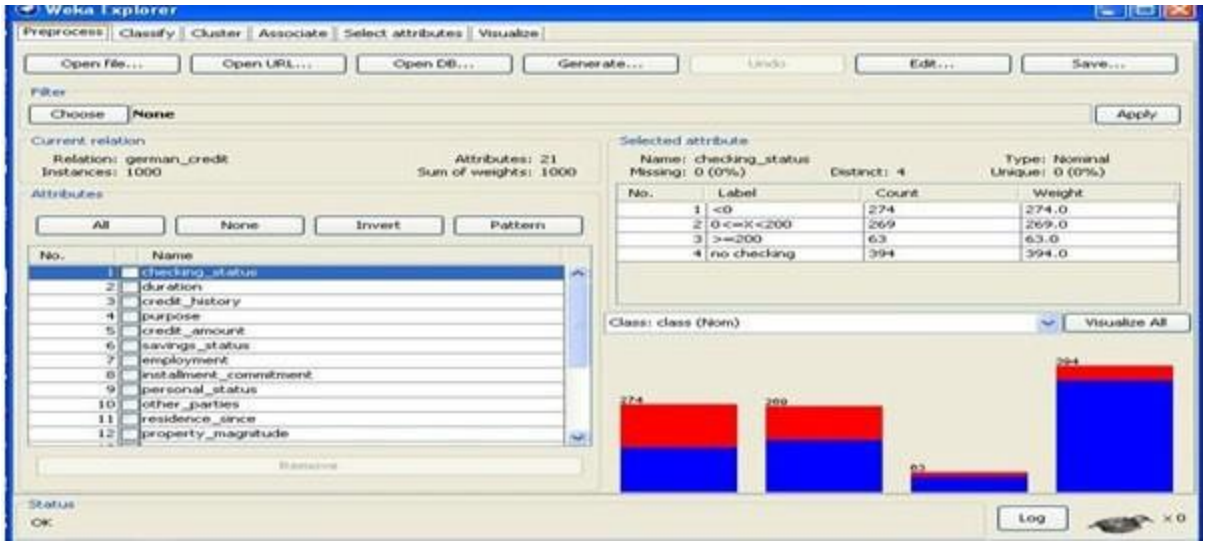
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such data set. Consisting of **1000** actual cases collected in Germany.

In spite of the fact that the data is German, you should probably make use of it for this assignment(Unless you really can consult a real loan officer!)

There are 20 attributes used in judging a loan applicant(ie., 7 Numerical attributes and 13 Categoricl or Nominal attributes). The goal is the classify the applicant into one of two categories. Good or Bad.

The total number of attributes present in German credit data are.

1. Checking_Status
2. Duration
3. Credit_history
4. Purpose
5. Credit_amout
6. Savings_status
7. Employment
8. Installment_Commitment
9. Personal_status
10. Other_parties
11. Residence_since
12. Property_Magnitude
13. Age
14. Other_payment_plans
15. Housing
16. Existing_credits
17. Job
18. Num_dependents
19. Own_telephone
20. Foreign_worker
21. Class



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

Steps for identifying categorical attributes

1. Double click on credit-g.arff file.
2. Select all categorical attributes.
3. Click on invert.
4. Then we get all real valued attributes selected
5. Click on remove
6. Click on visualize all.

Steps for identifying real valued attributes

1. Double click on credit-g.arff file.
2. Select all real valued attributes.
3. Click on invert.
4. Then we get all categorial attributes selected
5. Click on remove
6. Click on visualize all.

The following are the Categorical (or Nominal) attributes)

1. Checking_Status
2. Credit_history
3. Purpose
4. Savings_status
5. Employment
6. Personal_status
7. Other_parties
8. Property_Magnitude
9. Other_payment_plans
10. Housing
11. Job
12. Own_telephone
13. Foreign_worker

The following are the Numerical attributes.

1. Duration
2. Credit_amout
3. Installment_Commitment
4. Residence_since
5. Age
6. Existing_credits
7. Num_dependents

EXERCISE:8

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

EXERCISE:9

Explain One type of model that you can create is a Decision tree . train a Decision tree using the complete data set as the training data. Report the model obtained after training

EXERCISE :10

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

Ans) Steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on use training set in test options.
6. Click on start button.
7. On right side we find confusion matrix
8. Note the correctly classified instances.

Output:

If we used our above model trained on the complete dataset and classified credit as good/bad for each of the examples in that dataset. We can not get 100% training accuracy only **85.5%** of examples, we can classify correctly.

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

Ans)It is not good idea by using 100% training data set.

SIGNATURE OF FACULTY:

WEEK-6

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

Ans) steps followed are:

9. Double click on credit-g.arff file.
10. Click on classify tab.
11. Click on choose button.
12. Expand tree folder and select J48
13. Click on cross validations in test options.
14. Select folds as 10
15. Click on start
16. Change the folds to 5
17. Again click on start
18. Change the folds with 2
19. Click on start.
20. Right click on blue bar under result list and go to visualize tree

Output:

Cross-Validation Definition: The classifier is evaluated by cross validation using the number of folds that are entered in the folds text field.

In Classify Tab, Select cross-validation option and folds size is 2 then Press Start Button, next time change as folds size is 5 then press start, and next time change as folds size is 10 then press start.

SIGNATURE OF FACULTY:

WEEK:7

Check to see if the data shows a bias against "foreign workers" or "personal-status". One way to do this is to remove these attributes from the data set and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. Did removing these attributes have any significantly effect? Discuss.

Ans) steps followed are:

21. Double click on credit-g.arff file.
22. Click on classify tab.
23. Click on choose button.
24. Expand tree folder and select J48
25. Click on cross validations in test options.
26. Select folds as 10
27. Click on start
28. Click on visualization
29. Now click on preprocessor tab
30. Select 9th and 20th attribute
31. Click on remove button
32. Goto classify tab
33. Choose J48 tree
34. Select cross validation with 10 folds
35. Click on start button
36. Right click on blue bar under the result list and go to visualize tree.

Output:

We use the **Preprocess Tab in Weka GUI Explorer to remove an attribute "Foreign• workers" & "Perosnal_status" one by one.** In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

SIGNATURE OF FACULTY:

WEEK :8

Another question might be, do you really need to input so many attributes to get good results? May be only a few would do. For example, you could try just having attributes 2,3,5,7,10,17 and 21. Try out some combinations.(You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)

Ans) steps followed are:

- Double click on credit-g.arff file.
- Select 2,3,5,7,10,17,21 and tick the check boxes.
- Click on invert
- Click on remove
- Click on classify tab
- Choose trace and then algorithm as J48
- Select cross validation folds as 2
- Click on start.

OUTPUT:

We use the **Preprocess Tab** in Weka GUI Explorer to remove 2nd attribute (Duration). In Classify Tab, Select Use Training set option then Press Start Button, If these attributes removed from the dataset, we can see change in the accuracy compare to full data set when we removed.

SIGNATURE OF FACULTY:

WEEK-9

Sometimes, The cost of rejecting an applicant who actually has good credit might be higher than accepting an applicant who has bad credit. Instead of counting the misclassification equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. By using a cost matrix in weak. Train your decision tree and report the Decision Tree and cross validation results. Are they significantly different from results obtained in problem 6.

Ans) steps followed are:

1. Double click on credit-g.arff file.
2. Click on classify tab.
3. Click on choose button.
4. Expand tree folder and select J48
5. Click on start
6. Note down the accuracy values
7. Now click on credit arff file
8. Click on attributes 2,3,5,7,10,17,21
9. Click on invert
10. Click on classify tab
11. Choose J48 algorithm
12. Select Cross validation fold as 2
13. Click on start and note down the accuracy values.
14. Again make cross validation folds as 10 and note down the accuracy values.
15. Again make cross validation folds as 20 and note down the accuracy values.

OUTPUT:

In Weka GUI Explorer, Select Classify Tab, In that Select **Use Training set** option . In Classify Tab then press **Choose** button in that select J48 as Decision Tree Technique. In Classify Tab then press **More options** button then we get classifier evaluation options window in that select cost sensitive evaluation the press set option Button then we get Cost Matrix Editor. In that change classes as 2 then press Resize button. Then we get 2X2 Cost matrix. In Cost Matrix (0,1) location value change as 5, then we get modified cost matrix is as follows.

0.0 5.0
1.0 0.0

Then close the cost matrix editor, then press ok button. Then press start button.

SIGNATURE OF FACULTY:

WEEK:10

Do you think it is a good idea to prefect simple decision trees instead of having long complex decision tress? How does the complexity of a Decision Tree relate to the bias of the model?

Ans)

steps followed are:-
1)click on credit arff file
2)Select all attributes
3)click on classify tab
4)click on choose and select J48 algorithm
5)select cross validation folds with 2
6)click on start
7)Write down the time complexity Value.

OUTPUT:

SIGNATURE OF FACULTY:

WEEK : 11

You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning. Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross validation and report the Decision Trees you obtain? Also Report your accuracy using the pruned model Does your Accuracy increase?

Ans)

steps followed are:-

1)click on credit arff file
2)Select all attributes
3)click on classify tab
4)click on choose and select REP algorithm
5)select cross validation 2
6) click on start
7) Note down the results

OUTPUT:

SIGNATURE OF FACULTY:

WEEK :12

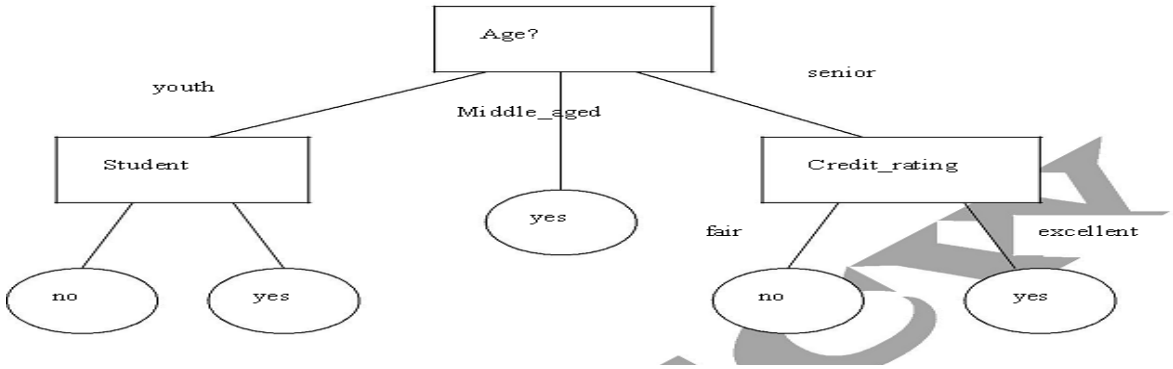
How Can you Convert Decision Tree in to "If then else Rules".Make Up your own Small Decision Tree consisting 2-3 levels and convert into a set of rules. There also exist different classifiers that output the model in the form of rules. One such classifier in weka is rules. PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one ! Can you predict what attribute that might be in this data set? OneR classifier uses a single attribute to make decisions(it chooses the attribute based on minimum error).Report the rule obtained by training a one R classifier. Rank the performance of j48,PART,oneR.

Ans)

Steps For Analyze Decision Tree:

1)click on credit arff file
2)Select all attributes
3) click on classify tab
4) click on choose and select J48 algorithm
5)select cross validation folds with 2
6)click on start
7) note down the accuracy value
8) again goto choose tab and select PART
9)select cross validation folds with 2
10)click on start
11) note down accuracy value
12) again goto choose tab and select One R
13)select cross validation folds with 2
14)click on start
15)note down the accuracy value.

Sample Decision Tree of Level 2-3.



OUTPUT:

SIGNATURE OF FACULTY:

Simple Project on Data Preprocessing

Data Preprocessing

Objective: Understanding the purpose of unsupervised attribute/instance filters for preprocessing the input data.

Follow the steps mentioned below to configure and apply a filter.

The preprocess section allows filters to be defined that transform the data in various ways. The Filter box is used to set up filters that are required. At the left of the Filter box is a Choose button. By clicking this button it is possible to select one of the filters in Weka. Once a filter has been selected, its name and options are shown in the field next to the Choose button. Clicking on this box brings up a GenericObjectEditor dialog box, which lets you configure a filter. Once you are happy with the settings you have chosen, click OK to return to the main Explorer window.

Now you can apply it to the data by pressing the Apply button at the right end of the Filter panel. The Preprocess panel will then show the transformed data. The change can be undone using the Undo button. Use the Edit button to view your transformed data in the dataset editor.

Try each of the following **Unsupervised Attribute Filters**.
(Choose -> weka -> filters -> unsupervised -> attribute)

- Use **ReplaceMissingValues**to replace missing values in the given dataset.
- Use the filter **Add** to add the attribute Average.
- Use the filter **AddExpression** and add an attribute which is the average of attributes M1 and M2. Name this attribute as AVG.
- Understand the purpose of the attribute filter **Copy.**
- Use the attribute filters **Discretize** and **PKIDiscretize** to discretize the M1 and M2 attributes into five bins. (NOTE: Open the file afresh to apply the second filter since there would be no numeric attribute to dicretize after you have applied the first filter.)
- Perform **Normalize** and **Standardize** on the dataset and identify the difference between

these operations.

- Use the attribute filter **FirstOrder** to convert the M1 and M2 attributes into a single attribute representing the first differences between them.
- Add a nominal attribute Grade and use the filter **MakeIndicator** to convert the attribute into a Boolean attribute.
- Try if you can accomplish the task in the previous step using the filter MergeTwoValues.
- Try the following transformation functions and identify the purpose of each
    - NumericTransform
    - NominalToBinary
    - NumericToBinary
    - Remove
    - RemoveType
    - RemoveUseless
    - ReplaceMissingValues
    - SwapValues

Try the following **Unsupervised Instance Filters.**

(Choose -> weka -> filters -> unsupervised -> instance)

- Perform **Randomize** on the given dataset and try to correlate the resultant sequence with the given one.
- Use **RemoveRange** filter to remove the last two instances.
- Use **RemovePercent** to remove 10 percent of the dataset.

- Apply the filter **RemoveWithValues** to a nominal and a numeric attribute