

DATA WAREHOUSING AND DATA MINING LAB (R17A0590)

LAB MANUAL AND RECORD

**B.TECH
(III YEAR – II SEM)
(2019-20)**



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 achieve high quality education in technical education that provides the skills and attitude to adapt to the global needs of the Information technology sector, through academic and research excellence.

Mission

- To equip the students with the cognizance for problem solving and to improve the teaching learning pedagogy by using innovative techniques.
- To strengthen the knowledge base of the faculty and students with the motivation towards possession of effective academic skills and relevant research experience.
- To promote the necessary moral and ethical values among the engineers ,for the betterment of the society.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1 – ANALYTICAL SKILLS

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

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

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 DATA MINING LAB

COURSE CODE: R17A0590

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	PO9	PO10	PO11
<ul style="list-style-type: none">• Ability to add mining algorithms as a component to the exiting tools.• Ability to apply mining techniques for realistic data.	√	√	√								√

DATAWARE HOUSE TOOLS

Cloudera	
Teradata	
Oracle	
TabLeau	

OPEN SOURCE DATA MINING TOOLS

WEKA	
Orange	
KNIME	
R-Programming	

DATA WAREHOUSING AND DATA MINING LAB

INDEX

S.No	Name of the Experiment	Pg No	Date	Signature
1	Data Processing Techniques: (i) Data Cleaning (ii) Data Transformation-Normalization (iii) Data Integration	1		
2	Data Warehouse Schemas: Star, Snowflake, Fact Constellation	12		
3	Data Cube Construction-OLAP operations	21		
4	Data Extraction, Transformations, Loading operations	31		
5	Implementation of Apriori algorithm	45		
6	Implementation of FP-Growth algorithm	54		
7	Implementation of Decision Tree Induction	62		
8	Calculating information gain measures	69		
9	Classification of data using Bayesian approach	77		
10	Classification of data using K-Nearest Neighbor approach	85		
11	Implementation of K-Means algorithm	92		

Experiment 1: Perform data preprocessing tasks

Preprocess Tab

1. Loading Data

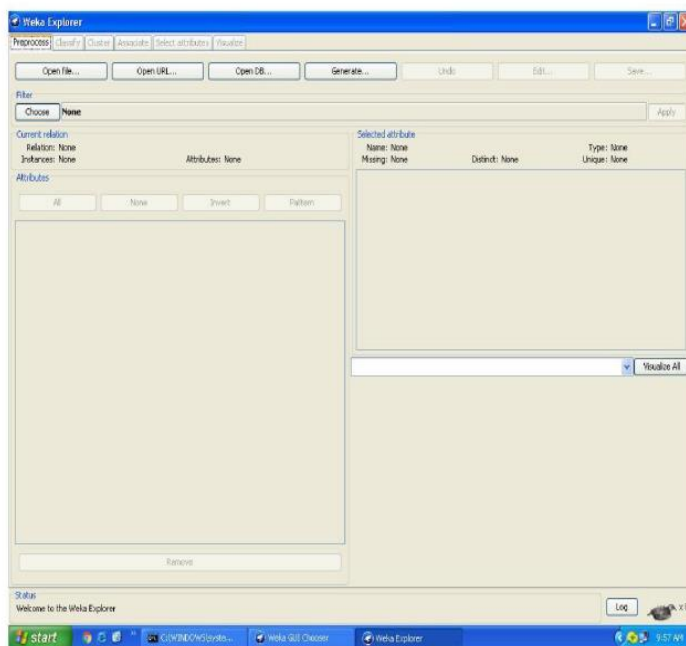
The first four buttons at the top of the preprocess section enable you to load data into WEKA:

1. Open fileBrings 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 DBReads data from a database. (Note that to make this work you might have to edit the file in weka/experiment/DatabaseUtils.props.)

4. GenerateEnables 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 information. 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 color-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.

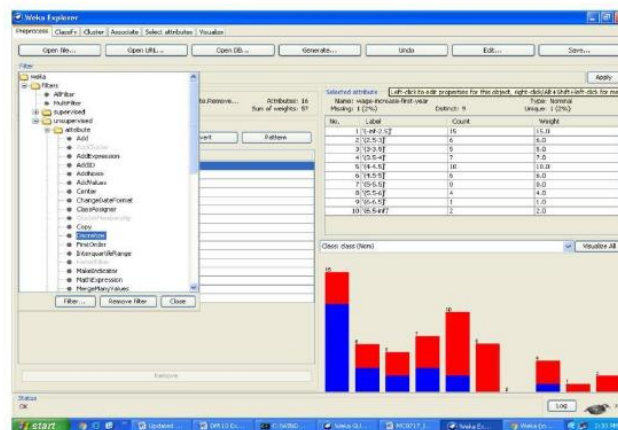
PREPROCESSING

- 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. 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 clipboard** 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.

Steps for run preprocessing tab in WEKA:

1. Open WEKA Tool.
2. Click on WEKA Explorer.
3. Click on Preprocessing tab button.

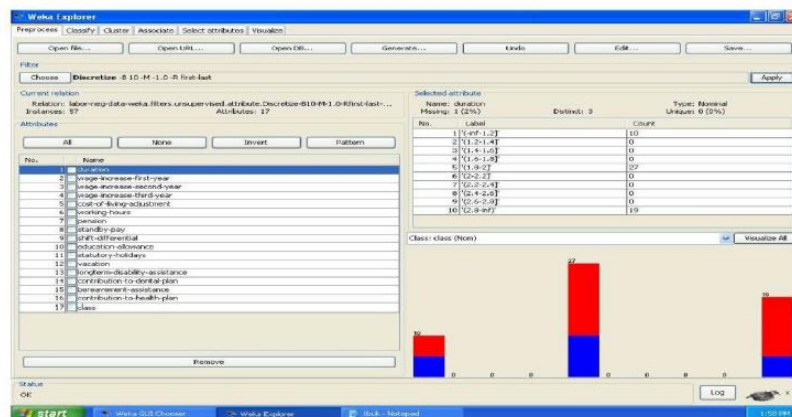
4. Click on open file button.
5. Choose WEKA folder in C drive.
6. Select and Click on data option button.
7. Choose labor data set and open file.
8. Choose filter button and select the Unsupervised-Discretize option and apply

Dataset labor.arff

The screenshot shows a file explorer window with the following columns: Name, Size, Date modified, Date created, Type, and Content type. The file 'labor.arff' is highlighted in the list.

Name	Size	Date modified	Date created	Type	Content type
labor.arff	1.0 KB	11/11/2017	11/11/2017	Text	text/plain

The following screenshot shows the effect of discretization



Exercise 1:

- a. Remove spaces from the file by using Java.
- b. Normalize the data using min-max normalization

RECORD NOTES

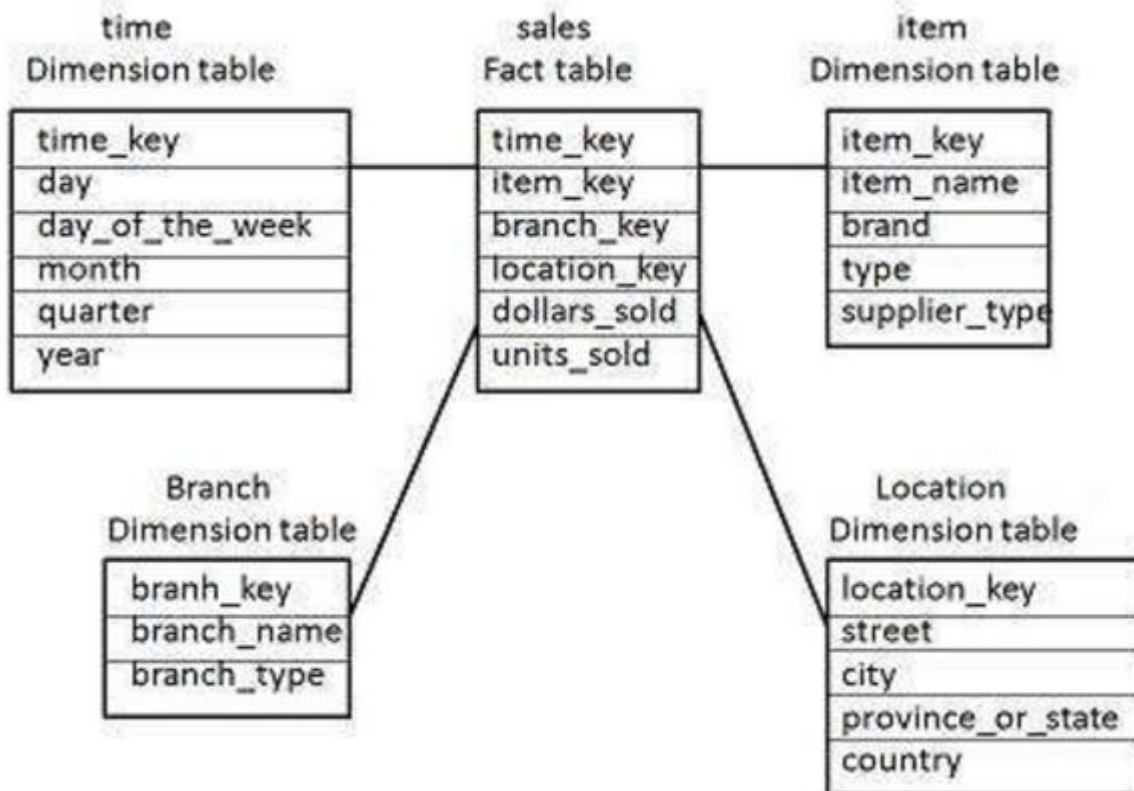
Experiment 2: Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).

Schema Definition

Multidimensional schema is defined using Data Mining Query Language (DMQL). The two primitives, cube definition and dimension definition, can be used for defining the data warehouses and data marts.

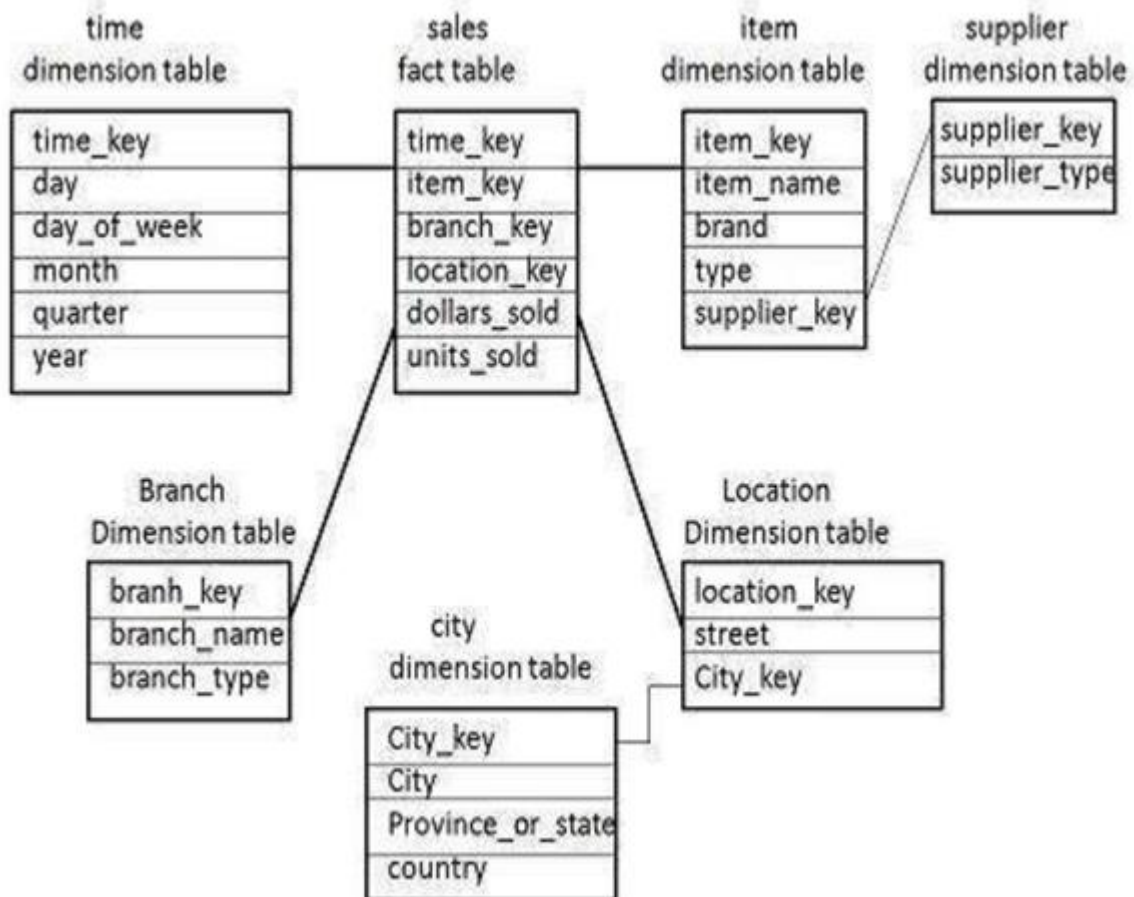
Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.
- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.
- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.



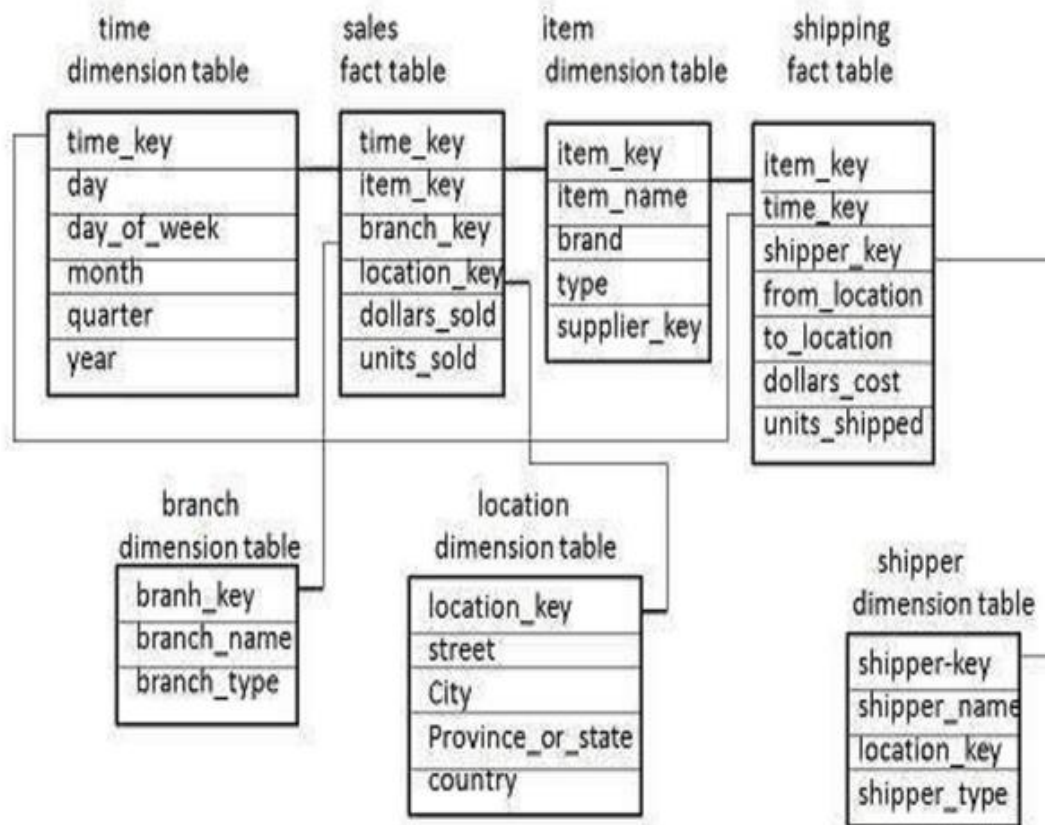
Snowflake Schema

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema is normalized. For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.
- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.



Fact Constellation Schema

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.
- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.



Exercise 2:

Design data warehouse schemas for Banking application.

RECORD NOTES

Experiment 3: Perform Various OLAP operations such slice, dice, roll up, drill up and pivot.

OLAP OPERATIONS

Online Analytical Processing Server (OLAP) is based on the multidimensional data model. It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information. Here is the list of OLAP operations:

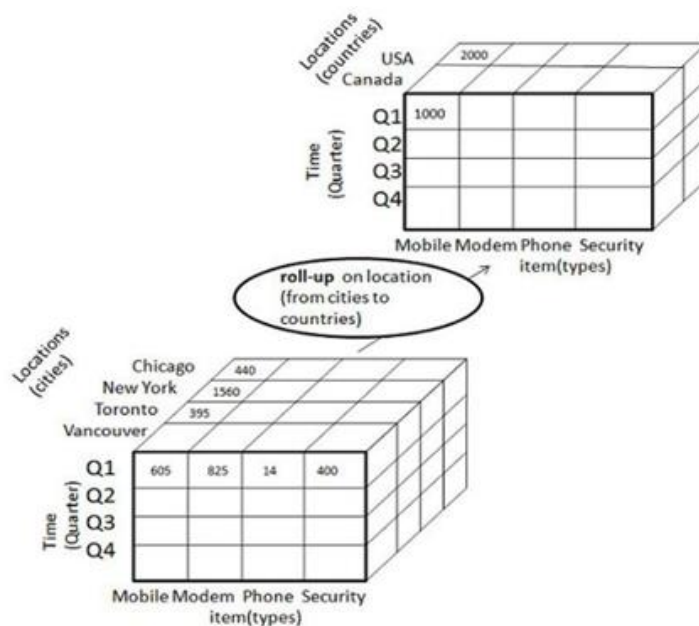
- ☐ Roll-up
- ☐ Drill-down
- ☐ Slice and dice
- ☐ Pivot (rotate)

Roll-up

Roll-up performs aggregation on a data cube in any of the following ways:

- ☐ By climbing up a concept hierarchy for a dimension
- ☐ By dimension reduction

The following diagram illustrates how roll-up works.



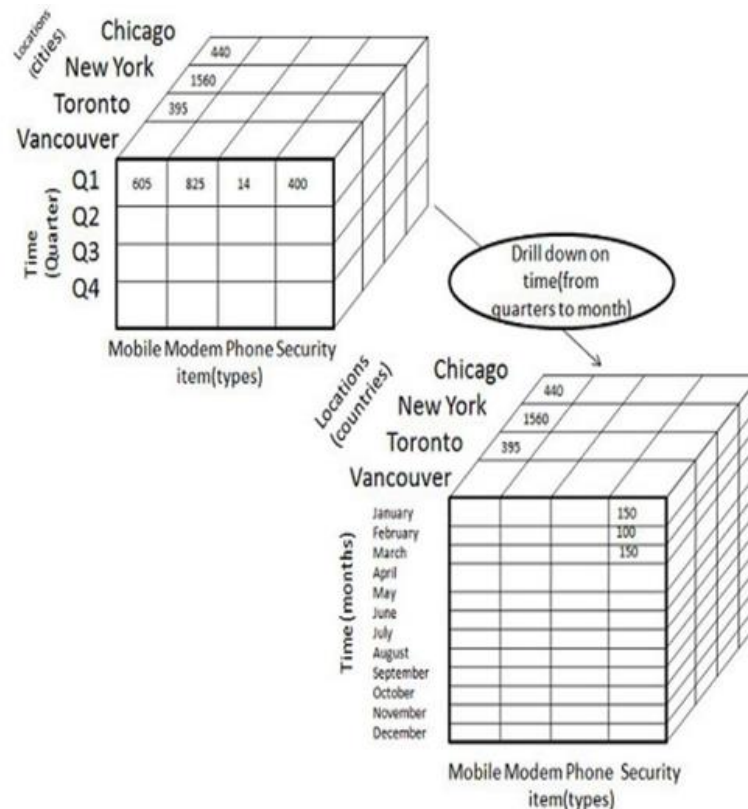
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by ascending the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or more dimensions from the data cube are removed.

Drill-down

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways:

- By stepping down a concept hierarchy for a dimension
- By introducing a new dimension.

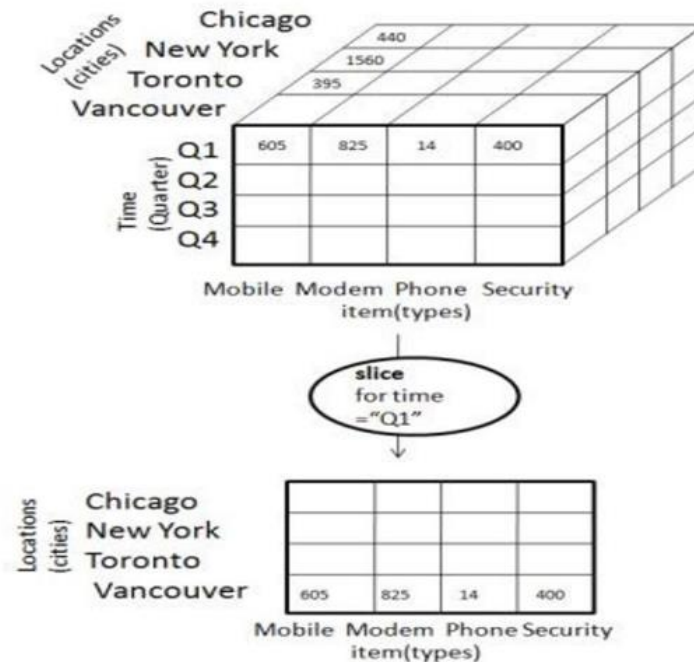
The following diagram illustrates how drill-down works:



- Drill-down is performed by stepping down a concept hierarchy for the dimension time.
- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is descended from the level of quarter to the level of month.
- When drill-down is performed, one or more dimensions from the data cube are added.
- It navigates the data from less detailed data to highly detailed data.

Slice

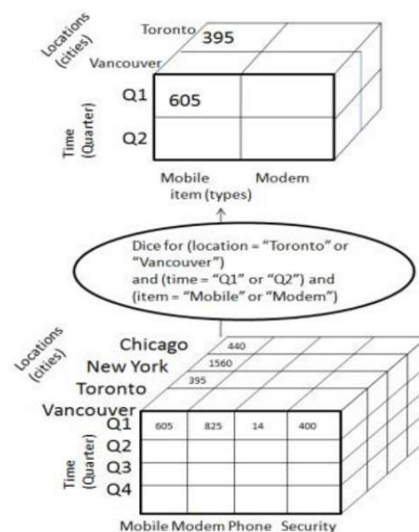
The slice operation selects one particular dimension from a given cube and provides a new sub-cube. Consider the following diagram that shows how slice works.



- Here Slice is performed for the dimension "time" using the criterion time = "Q1".
- It will form a new sub-cube by selecting one or more dimensions.

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

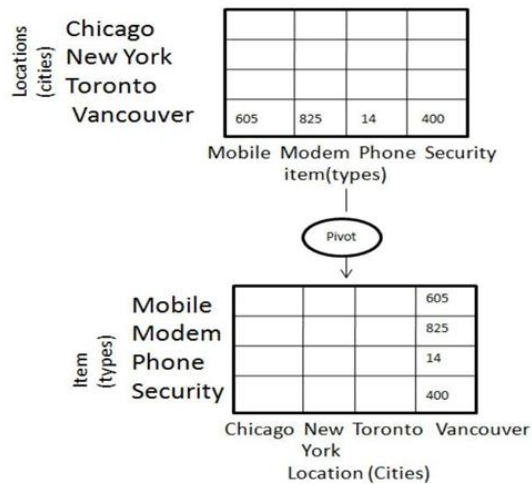


The dice operation on the cube based on the following selection criteria involves three dimensions.

- (location = "Toronto" or "Vancouver")
- (time = "Q1" or "Q2")
- (item = " Mobile" or "Modem")

Pivot

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



Exercise 3:

Apply the OLAP operations for the above banking application.

RECORD NOTES

Experiment 4: ETL scripts and implement using data warehouse tools.

ETL comes from Data Warehousing and stands for Extract-Transform-Load. ETL covers a process of how the data are loaded from the source system to the data warehouse. Extraction–transformation–loading (ETL) tools are pieces of software responsible for the extraction of data from several sources, its cleansing, customization, reformatting, integration, and insertion into a data warehouse.

Building the ETL process is potentially one of the biggest tasks of building a warehouse; it is complex, time consuming, and consumes most of data warehouse project's implementation efforts, costs, and resources.

Building a data warehouse requires focusing closely on understanding three main areas:

1. Source Area- The source area has standard models such as entity relationship diagram.
2. Destination Area- The destination area has standard models such as star schema.
3. Mapping Area- But the mapping area has not a standard model till now.

ETL Process:

Extract

The Extract step covers the data extraction from the source system and makes it accessible for further processing. The main objective of the extract step is to retrieve all the required data from the source system with as little resources as possible. The extract step should be designed in a way that it does not negatively affect the source system in terms of performance, response time or any kind of locking.

There are several ways to perform the extract:

- Update notification - if the source system is able to provide a notification that a record has been changed and describe the change, this is the easiest way to get the data.
- Incremental extract - some systems may not be able to provide notification that an update has occurred, but they are able to identify which records have been modified and provide an extract of such records. During further ETL steps, the system needs to identify changes and propagate it down. Note, that by using daily extract, we may not be able to handle deleted records properly.
- Full extract - some systems are not able to identify which data has been changed at all, so a full extract is the only way one can get the data out of the system. The full extract requires keeping a copy of the last extract in the same format in order to be able to identify changes. Full extract handles deletions as well.

Transform

The transform step applies a set of rules to transform the data from the source to the target. This includes converting any measured data to the same dimension (i.e. conformed dimension) using the same units so that they can later be joined. The transformation step also requires joining data from several sources, generating aggregates, generating surrogate keys, sorting, deriving new calculated values, and applying advanced validation rules.

Load

During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible. The target of the Load process is often a database. In order to make the load process efficient, it is helpful to disable any constraints and indexes before the load and enable them back only after the load completes. The referential integrity needs to be maintained by ETL tool to ensure consistency.

ETL method

ETL as scripts that can just be run on the database. These scripts must be re-runnable: they should be able to be run without modification to pick up any changes in the legacy data, and automatically work out how to merge the changes into the new schema.

In order to meet the requirements, my scripts must:

- 1.INSERT rows in the new tables based on any data in the source that hasn't already been created in the destination
- 2.UPDATE rows in the new tables based on any data in the source that has already been inserted in the destination
- 3.DELETE rows in the new tables where the source data has been deleted

Next step is to design the architecture for custom ETL solution.

- 1.create two new schemas on the new database: LEGACY and MIGRATE
- 2.take a snapshot of all data in the legacy database, and load it as tables in the LEGACY schema
- 3.grant read-only on all tables in LEGACY to MIGRATE
- 4.Grant CRUD on all tables in the target schema to MIGRATE.

WEKA

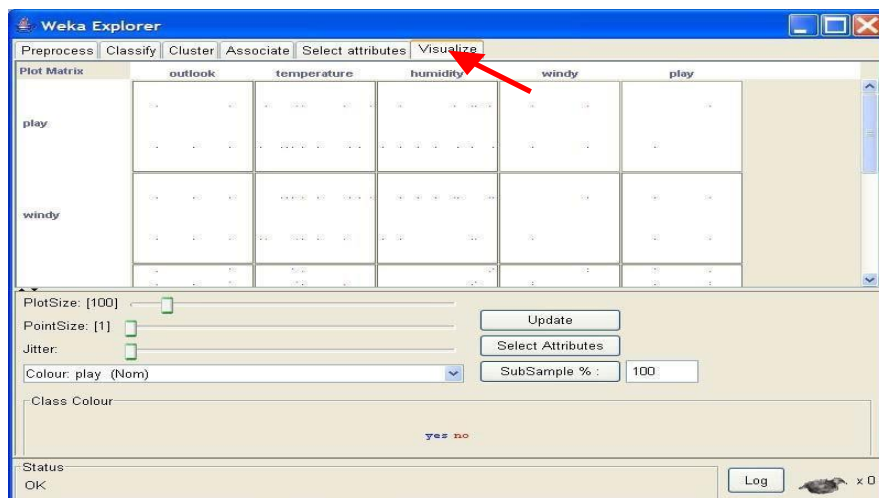
Visualization Features:

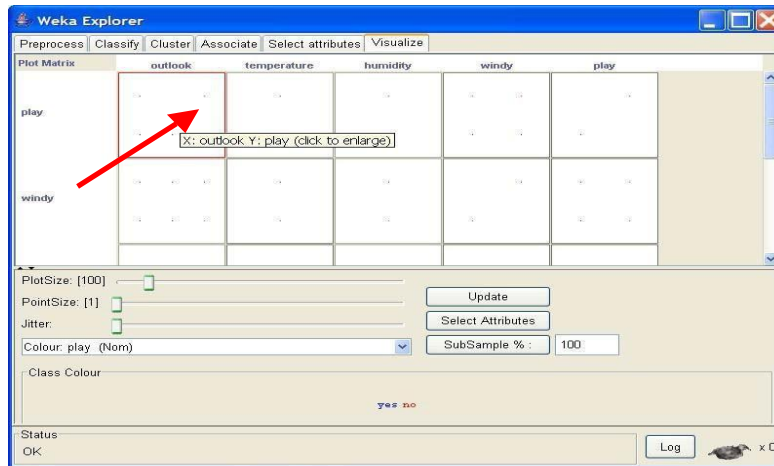
WEKA's visualization allows you to visualize a 2-D plot of the current working relation. Visualization is very useful in practice, it helps to determine difficulty of the learning problem. WEKA can visualize single attributes (1-d) and pairs of attributes (2-d), rotate 3-d visualizations (Xgobi-style). 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 colorpalette.

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.

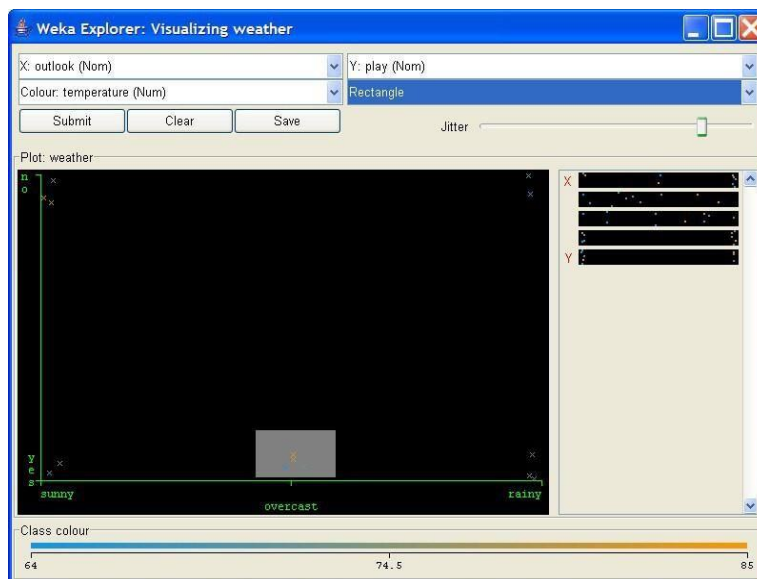
Selecting Instances

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]:

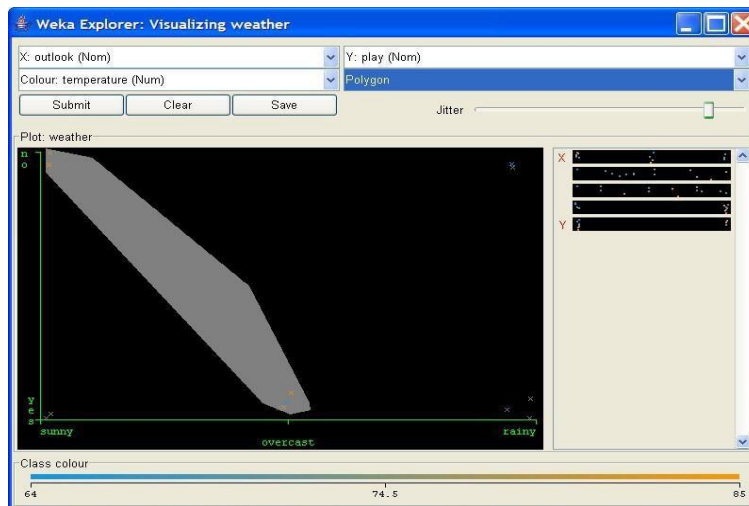
1. **Select Instance.** Click on an individual data point. It brings up a window listing attributes of the point. If more than one point will appear at the same location, more than one set of attributes will be shown.



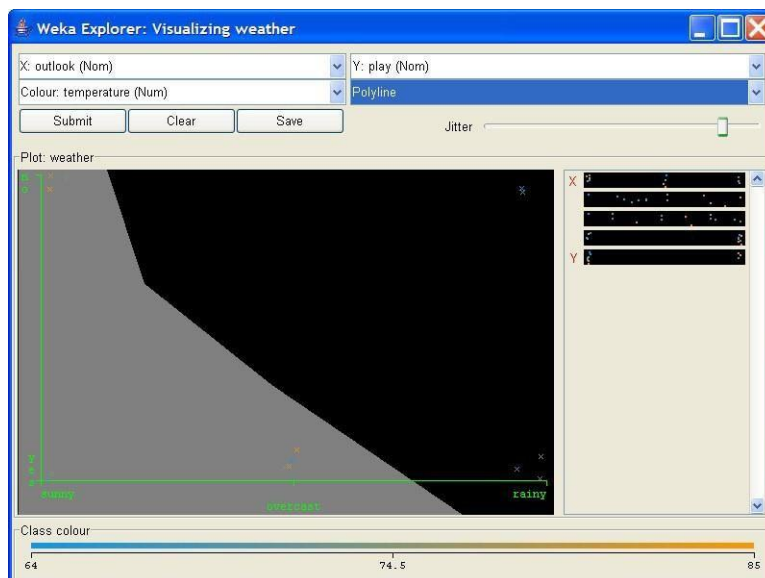
2. **Rectangle.** You can create a rectangle by dragging it around the point.



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.

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.



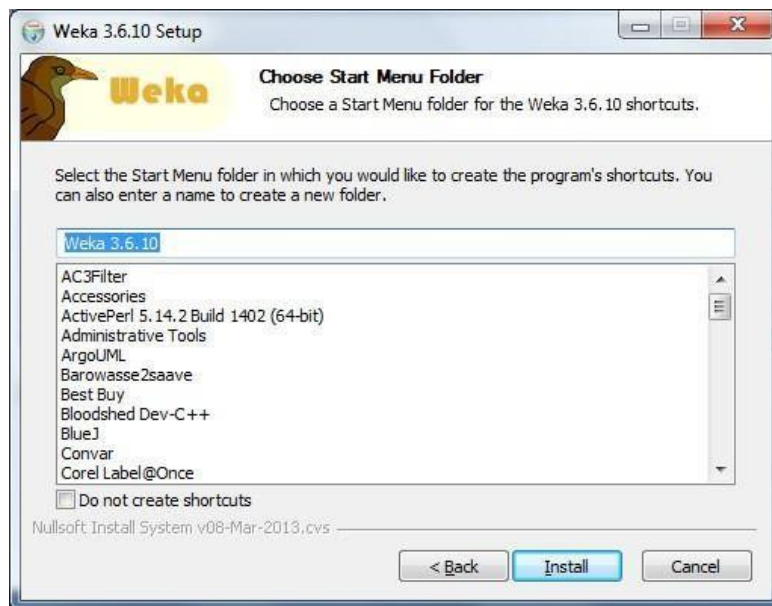
6. As your requirement do the necessary changes of settings and click Next. Full and Associate files are the recommended settings.



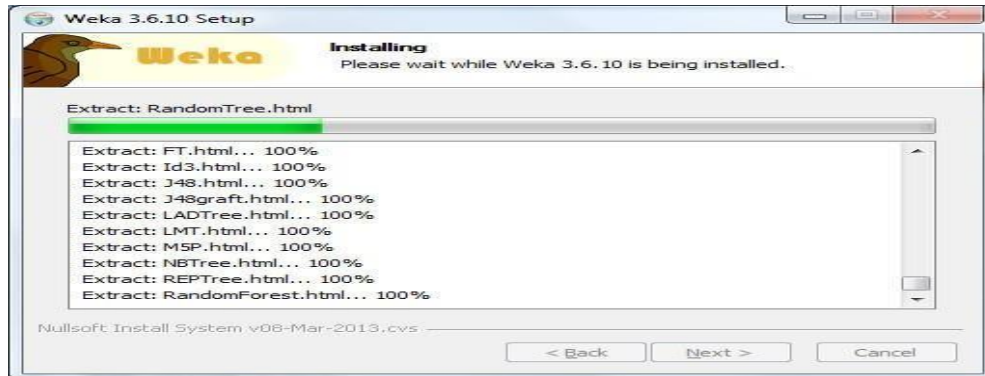
7. Change to your desire installation location.



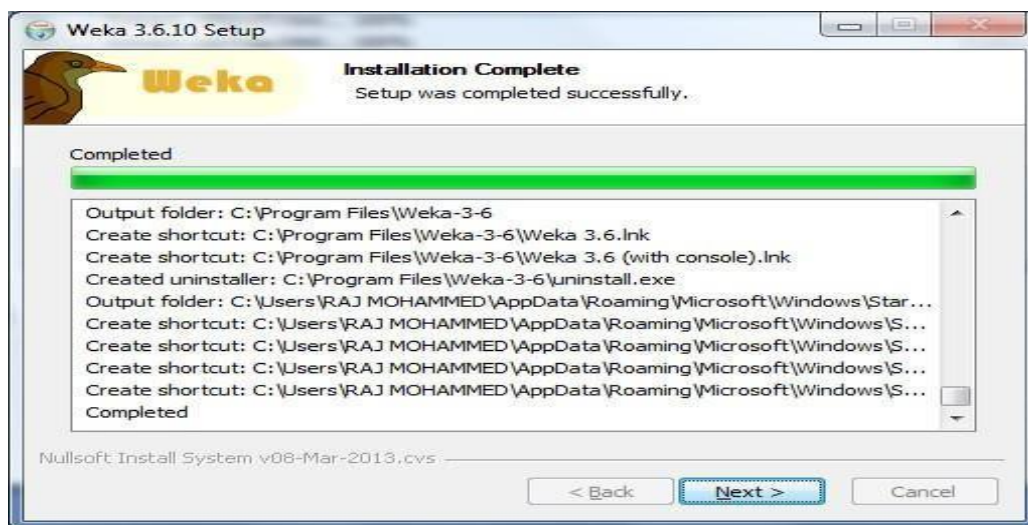
8. If you want a shortcut then check the box and click Install.



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

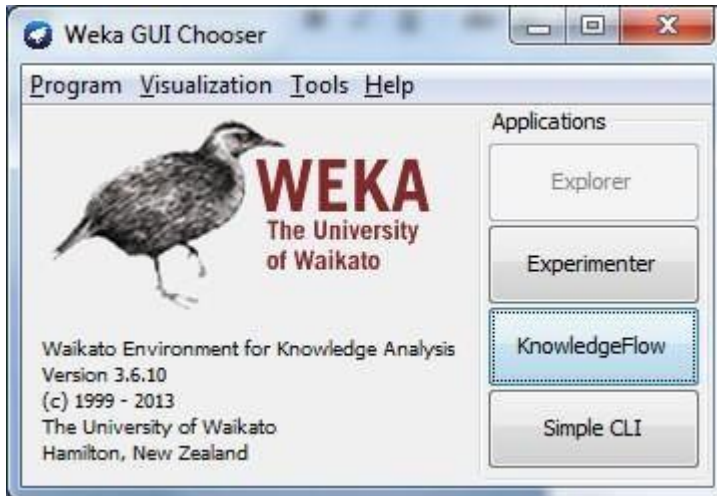


10. After complete installation click on Next.



11. Click on the Finish and start Mining.





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

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

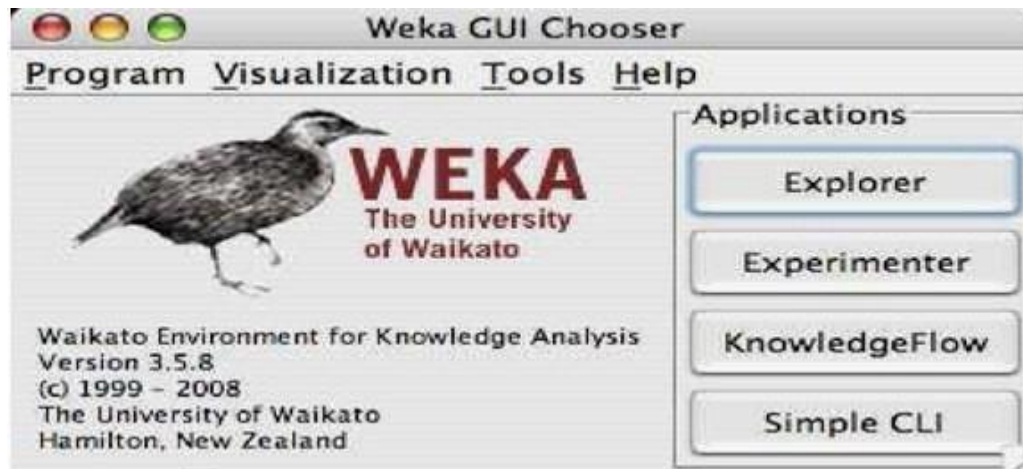
WEKA

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 of open 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

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.

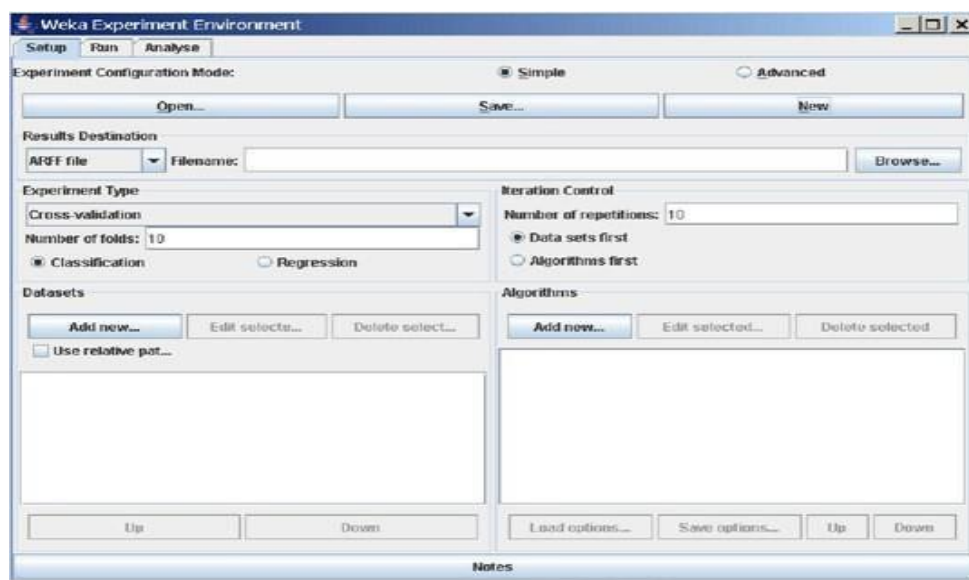
5. **Select attributes.** Select the most relevant attributes in the data.

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.

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.

Experiment 5: Implementation of Apriori algorithm

Association rule mining is defined as: Let I be a set of n binary attributes called items. Let D 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 $X \Rightarrow Y$ where $X, Y \subseteq I$ and $X \cap Y = \Phi$. 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 $I = \{\text{milk, bread, butter, beer}\}$ 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.

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 $\text{supp}(X)$ 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 $\{\text{milk, bread}\}$ 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 L_k , $L_k = \{I_1, I_2, \dots, I_k\}$, association rules with this itemsets are generated in the following way: the first rule is $\{I_1, I_2, \dots, I_{k-1}\} \Rightarrow \{I_k\}$, 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 L_{k-1} .

- Join Step.
- C_k is generated by joining L_{k-1} with 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 C_k : Candidate itemset of size k

- (L_k : frequent itemset of size k) Apriori Pseudocode
- Apriori* (T, ϵ)

$L \leftarrow \{ \text{Large Itemsets that appear in more than transactions} \}$
 while $L_{(k-1)} \neq \emptyset$ $C(k) \leftarrow \text{Generate}(L_{k-1})$ for transactions $t \in T$
 $C(t) \text{Subset}(C_k, t)$

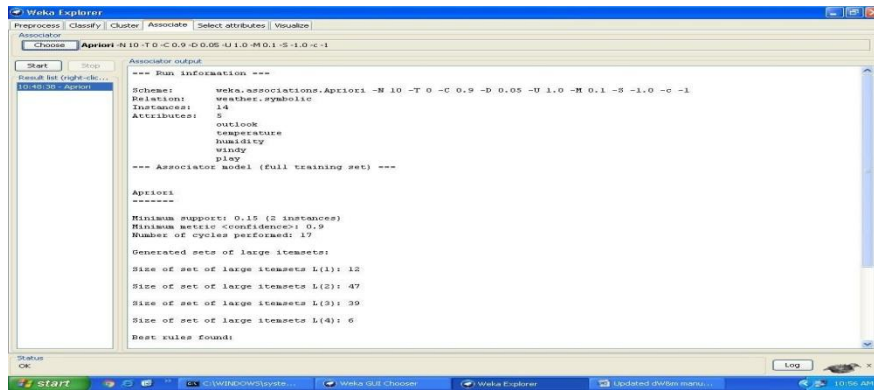
for candidates $c \in C(t)$

$\text{count}[c] \leftarrow \text{count}[c] + 1$ $L(k) \leftarrow \{ c$

$\in C(k) \mid \text{count}[c] \geq \epsilon K \}$ $K \leftarrow K + 1$ return $\bigcup L(k)$ k .

Steps for run Apriori algorithm 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 Weather data set and open file.
- o Click on Associate tab and Choose Apriori algorithm
- o Click on start button.



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,

$$support \square \frac{(X \sqcup Y).count}{n}$$

$$confidence \square \frac{(X \sqcup Y).count}{X.count}$$

support = support($\{A \cup C\}$)

confidence = support($\{A \cup C\}$)/support($\{A\}$)

Exercise 5:

Apply the Apriori algorithm on Airport noise monitoring data set, Discriminating between patients with Parkinsons and neurological diseases using voice recordings dataset.

[<https://archive.ics.uci.edu/ml/machine-learning-databases/00000/> refer this link for datasets]

RECORD NOTES

Experiment 6: Implementation of FP-Growth algorithm

By using the FP-Growth method, the number of scans of the entire database can be reduced to two. The algorithm extracts frequent item sets that can be used to extract association rules. This is done using the *support* of an item set. The main idea of the algorithm is to use a divide and conquer strategy:

Compress the database which provides the frequent sets; then divide this compressed database into a set of conditional databases, each associated with a frequent set and apply data mining on each database.

To compress the data source, a special data structure called the *FP-Tree* is needed [26]. The tree is used for the data mining part.

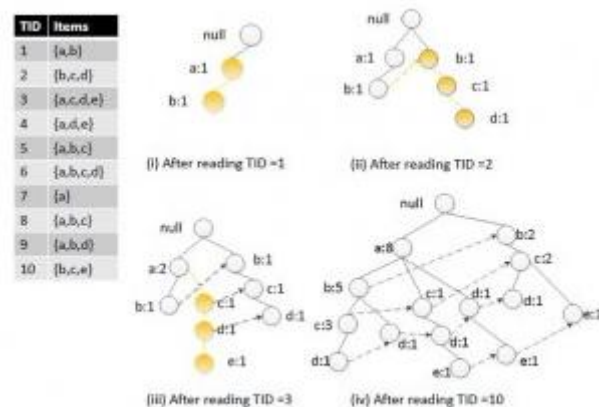
Finally the algorithm works in two steps:

1. Construction of the FP-Tree
2. Extract frequent item sets

Construction of the FP-Tree

The FP-Tree is a compressed representation of the input. While reading the data source each transaction t is mapped to a path in the FP-Tree. As different transaction can have several items in common, their path may overlap. With this it is possible to compress the structure.

The below figure shows an example for the generation of an FP-tree using 10 transactions.



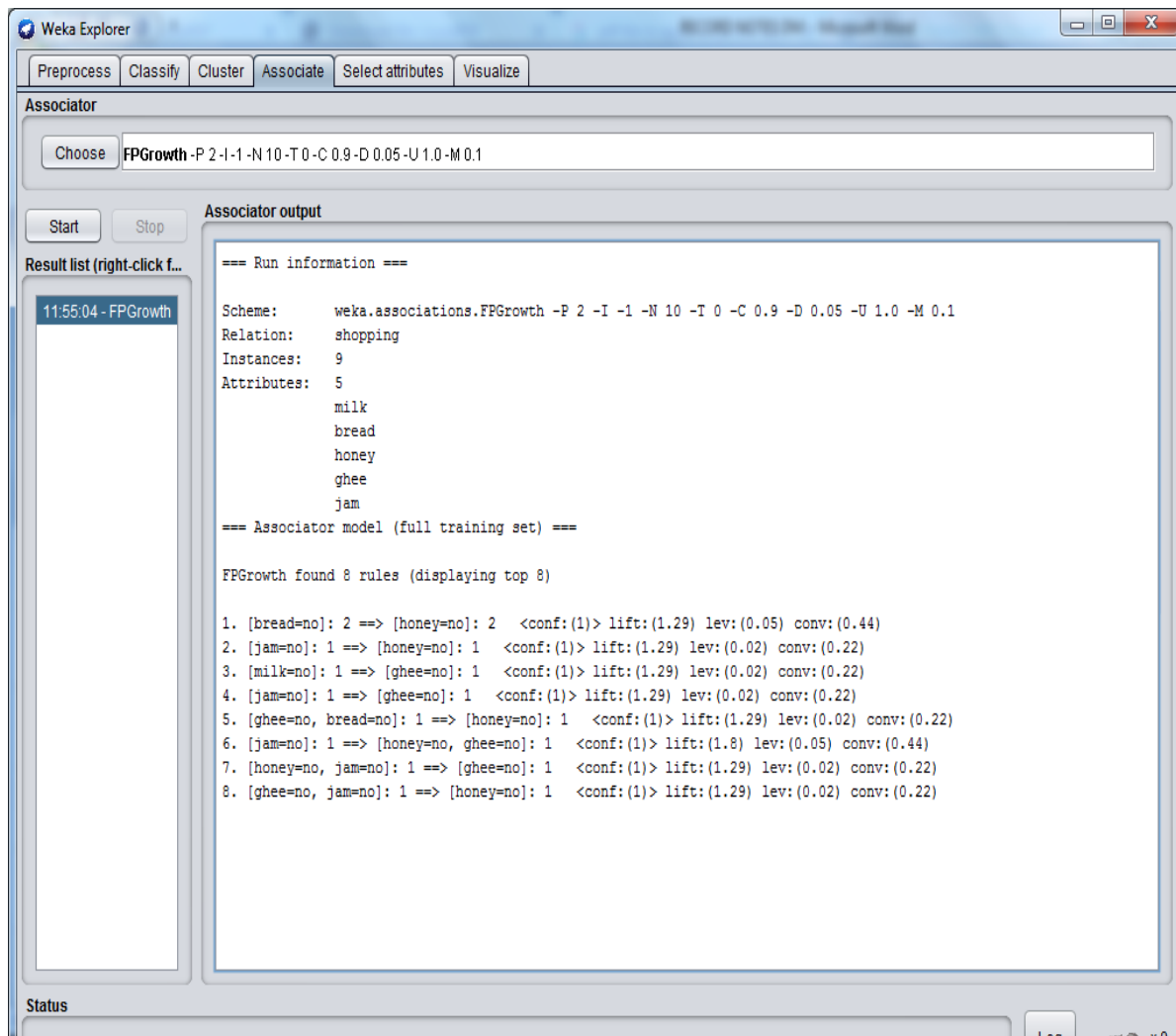
Extract frequent item sets

A bottom-up strategy starts with the leaves and moves up to the root using a divide and conquer strategy. Because every transaction is mapped on a path in the FP-Tree, it is possible to mine frequent item sets ending in a particular item.

Steps:

- 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 a data set and open file.
- Click on Associate tab and Choose FP-Growth algorithm
- Click on start button.



Exercise 6:

Apply FP-Growth algorithm on Blood Transfusion Service Center data set

RECORD NOTES

Experiment 7: Implementation of Decision Tree Induction

Steps to model decision tree.

1. Double click on credit-g.arff file.
2. Consider all the 21 attributes for making decision tree.
3. Click on classify tab.
4. Click on choose button.
5. Expand tree folder and select J48
6. Click on use training set in test options.
7. Click on start button.
8. Right click on result list and choose the visualize tree to get decision tree.

We created a decision tree by using J48 Technique for the complete dataset as the training data. The following model obtained after training.

The screenshot shows the Weka Explorer interface with the 'Classify' tab selected. The classifier chosen is 'J48 - C 0.25 - M 2'. The 'Test options' section has 'Cross-validation' selected with 'Folds' set to 10. The 'Classifier output' pane displays the following results:

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

Correctly Classified Instances      10       100 %
Incorrectly Classified Instances      0         0 %
Kappa statistic                      1
Mean absolute error                   0
Root mean squared error               0
Relative absolute error               0 %
Root relative squared error           0 %
Total Number of Instances           10

=== Detailed Accuracy By Class ===

      TP Rate  FP Rate  Precision  Recall  F-Measure  MCC   ROC Area  PRC Area  Class
      1.000   ?       1.000     1.000   1.000     ?     ?       1.000   Iris-setosa
      ?       0.000   ?         ?       ?         ?     ?       ?       Iris-versicolor
      ?       0.000   ?         ?       ?         ?     ?       ?       Iris-virginica
Weighted Avg.   1.000   ?       1.000     1.000   1.000     ?     ?       1.000

=== Confusion Matrix ===

  a  b  c  <-- classified as
10  0  0 | a = Iris-setosa
 0  0  0 | b = Iris-versicolor
 0  0  0 | c = Iris-virginica
```

The 'Result list' on the left shows two entries: '12:02:50 - trees_J48' and '12:03:04 - trees_J48', with the second entry selected.

Exercise 7:

Apply decision tree algorithm to book a table in a hotel/ book a train ticket/ movie ticket.

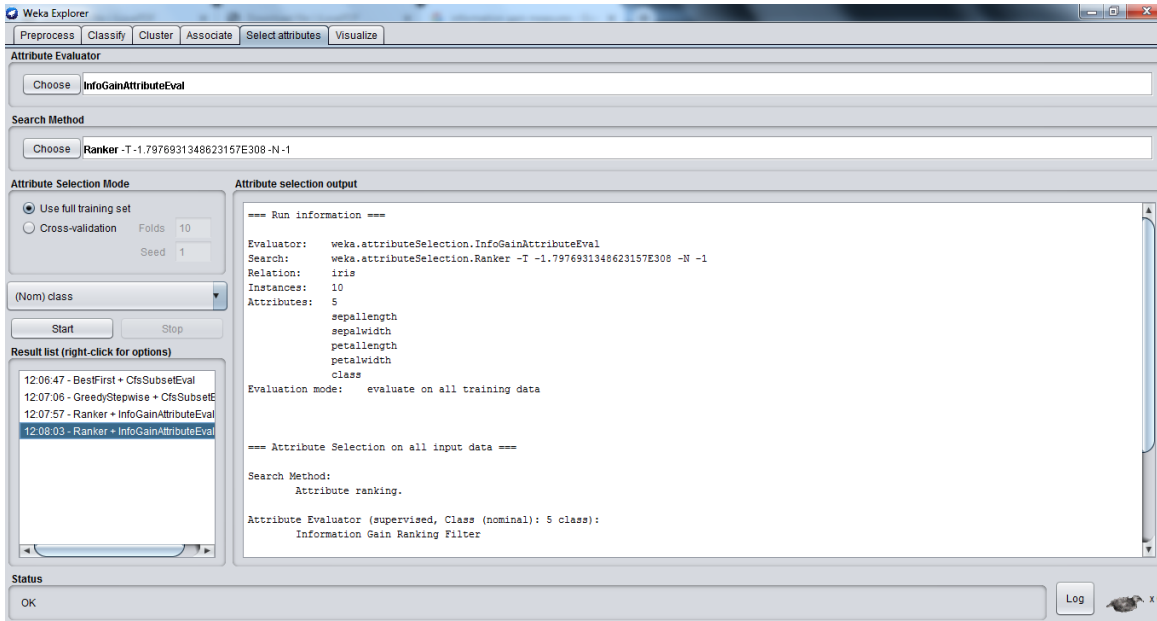
RECORD NOTES

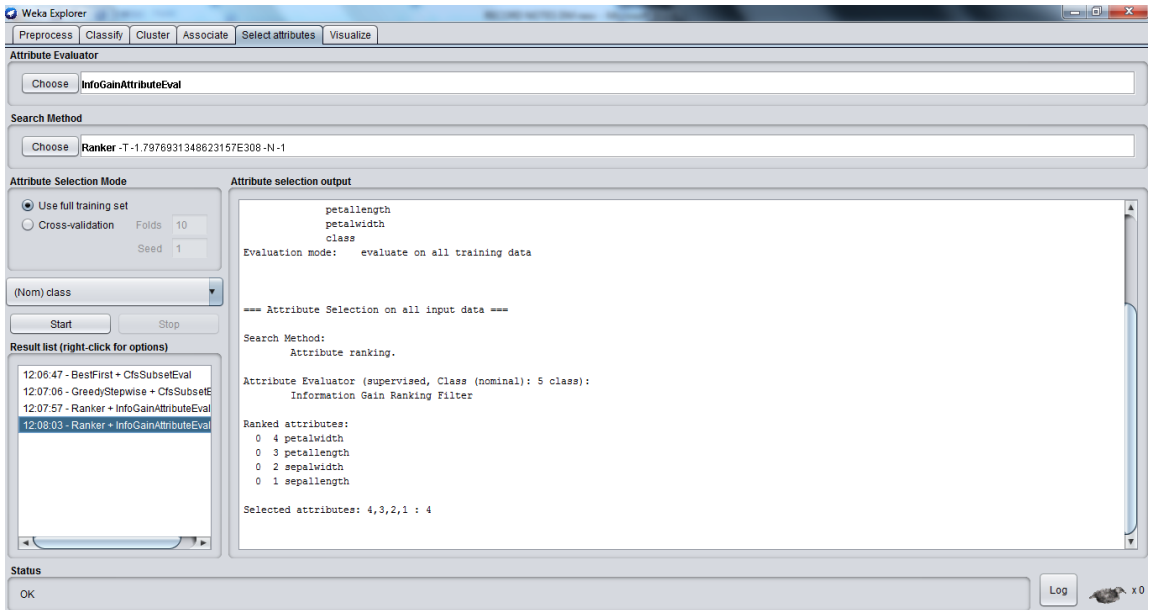
Experiment 8: calculating information gain measures.

Information gain (IG) measures how much “**information**” a feature gives us about the class. – Features that perfectly partition should give maximal **information**. – Unrelated features should give no **information**. It **measures** the reduction in entropy. CfsSubsetEval aims to identify a subset of attributes that are highly correlated with the target while not being strongly correlated with one another. It searches through the space of possible attribute subsets for the “best” one using the BestFirst search method by default, although other methods can be chosen. To use the wrapper method rather than a filter method, such as CfsSubsetEval, first select WrapperSubsetEval and then configure it by choosing a learning algorithm to apply and setting the number of cross-validation folds to use when evaluating it on each attribute subset.

Steps:

- Open WEKA Tool.
- Click on WEKA Explorer.
 - Click on Preprocessing tab button.
 - Click on open file button.
- Select and Click on data option button.
- Choose a data set and open file.
- Click on select attribute tab and Choose attribute evaluator, search method algorithm
- Click on start button.





Exercise 8:

Calculate the information gain on weather data set(for each attributes separately).

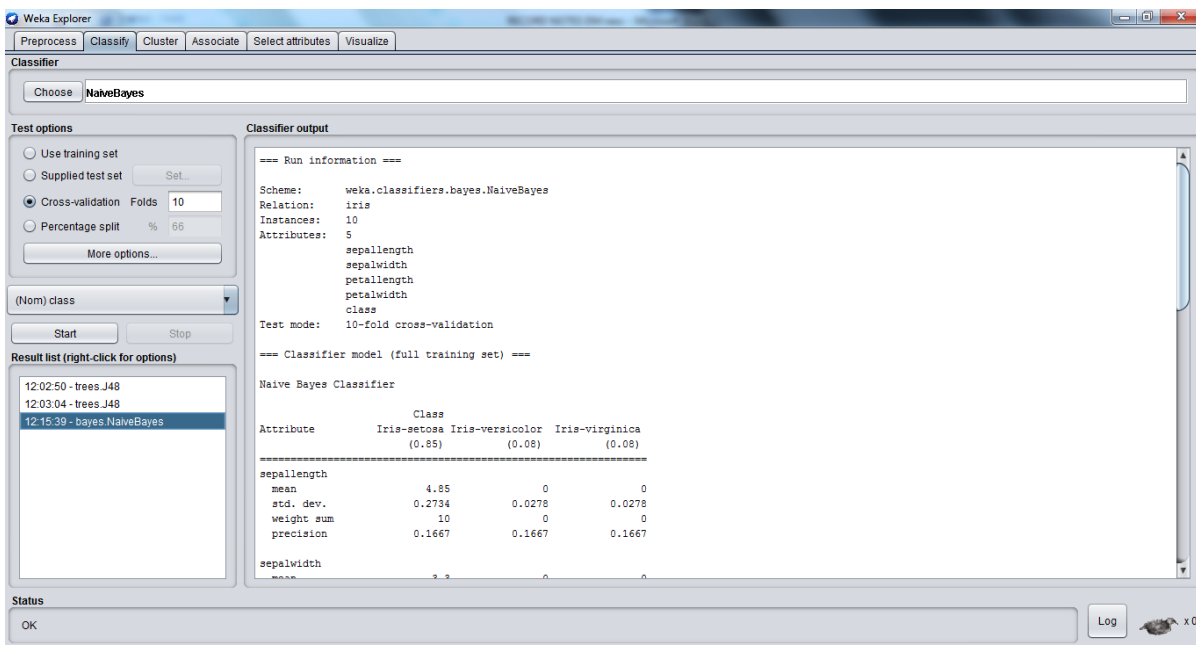
RECORD NOTES

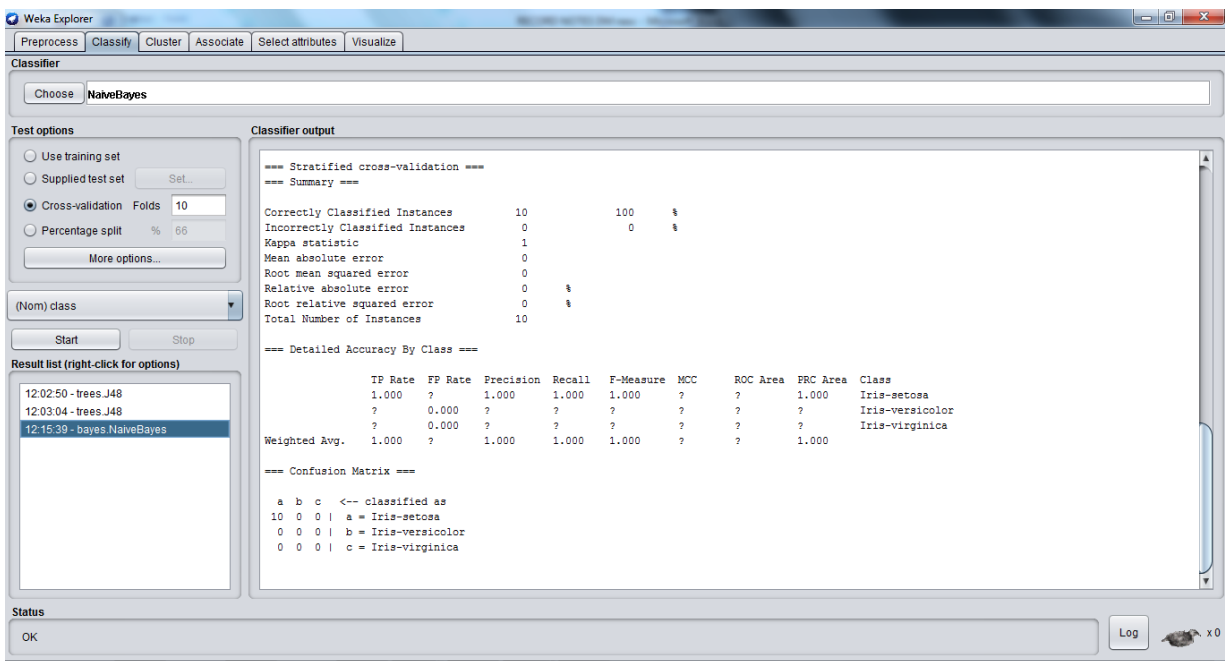
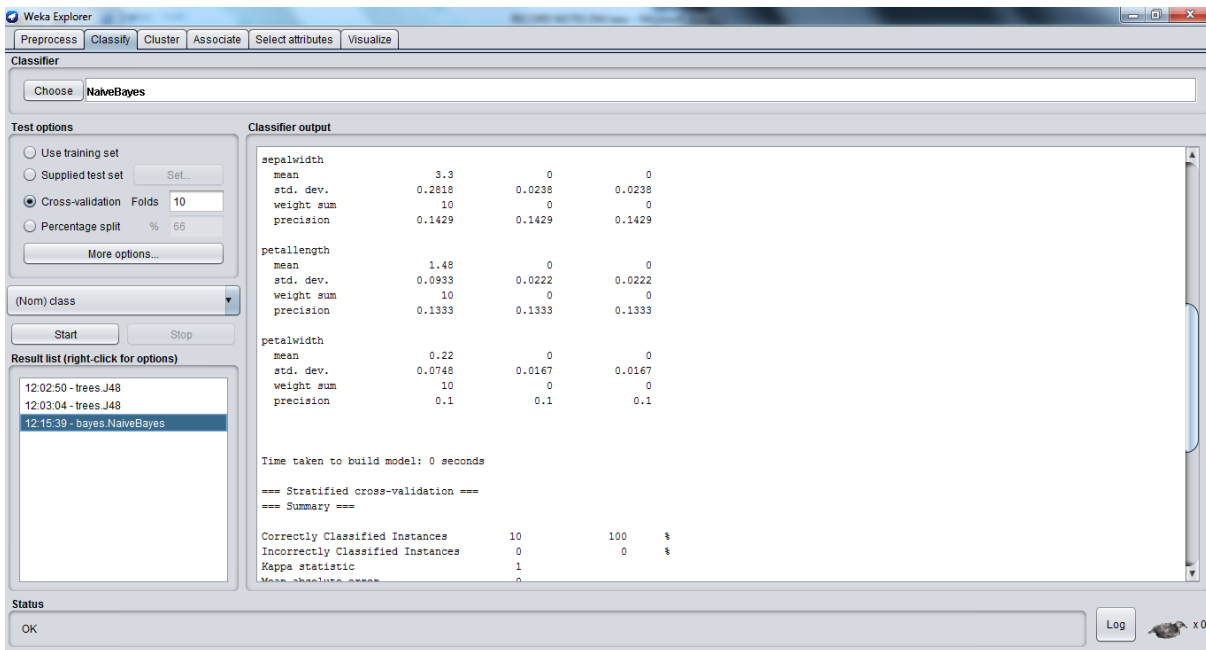
Experiment 9: classification of data using Bayesian approach

Bayesian classification is based on Bayes' Theorem. Bayesian classifiers are the statistical classifiers. Bayesian classifiers can predict class membership probabilities such as the probability that a given tuple belongs to a particular class.

Steps:

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 a data set and open file.
8. Click on classify tab and Choose Naïve-bayes algorithm and select use training set test option.'
9. Click on start button.





Exercise 9

Classify data (lung cancer/ diabetes /liver disorder)using Bayesian approach .

RECORD NOTES

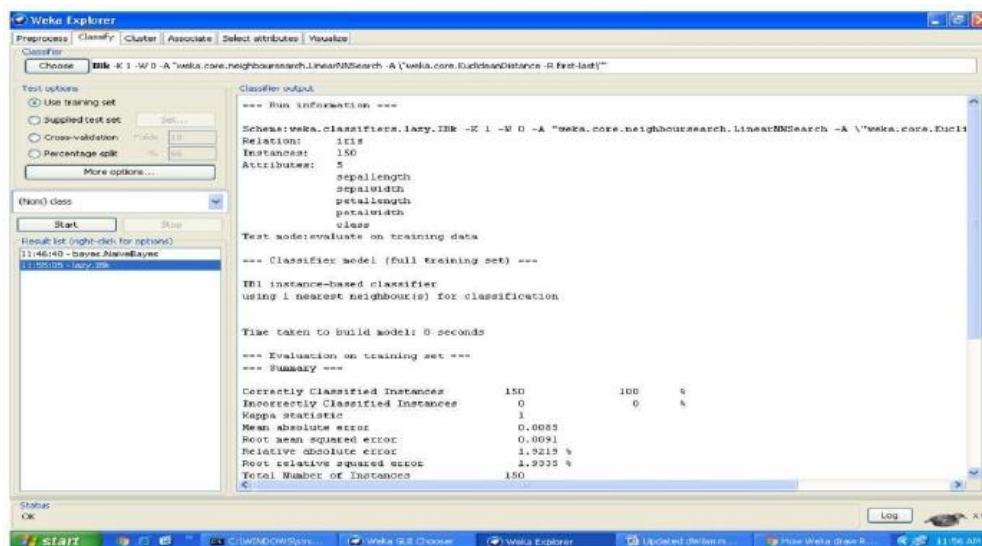
Experiment 10: classification of data using K-nearest neighbor approach

K nearest neighbors is a simple **algorithm** that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well –

- **Lazy learning algorithm** – KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- **Non-parametric learning algorithm** – KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Steps:

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 a data set and open file.
8. Click on classify tab and Choose k-nearest neighbor and select use training set test option.
9. Click on start button.



Exercise 10:

Perform analysis on iris data set and build cluster using K-nearest neighbor approach.

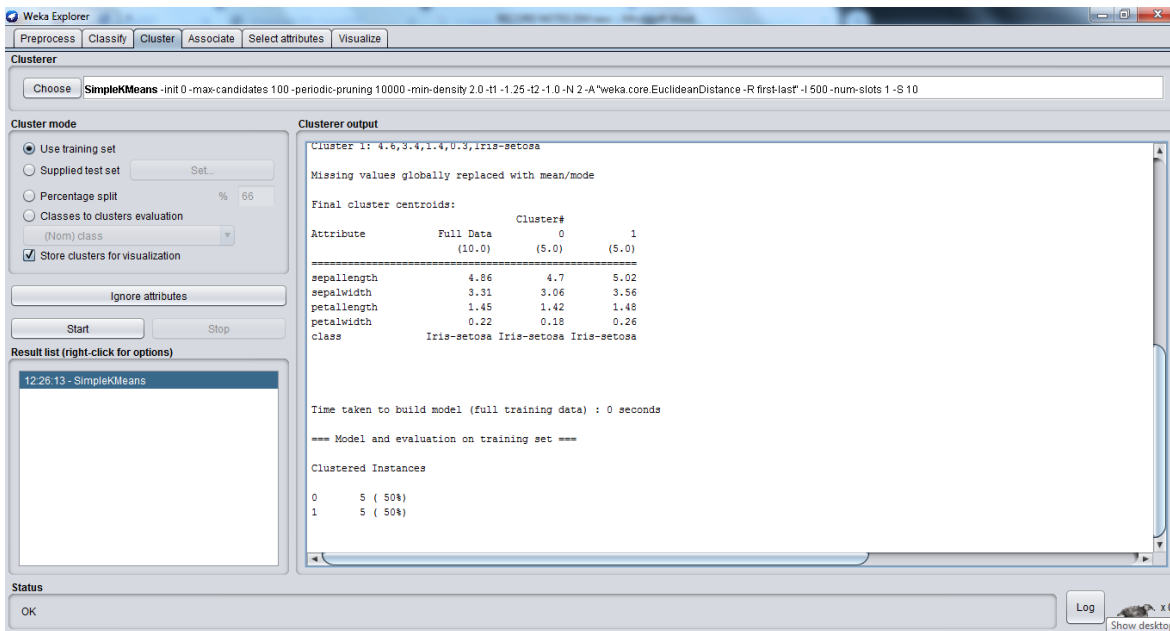
RECORD NOTES

Experiment 11: implementation of K-means algorithm

Kmeans algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to **only one group**. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster's centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

Steps:

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 a data set and open file.
8. Click on cluster tab and Choose k-means algorithm.
9. Click on start button.



Exercise 11:

Implement of K-means clustering using crime dataset.

RECORD NOTES

