DATA WAREHOUSING AND DATA MINING LAB (R20A1283)

LAB MANUAL AND RECORD

B.TECH
(III YEAR – II SEM)
(2022-23)

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 – PROFESSIONALISM &CITIZENSHIP

To create and sustain a community of learning in which students acquire knowledge and learn to apply it professionally with due consideration for ethical, ecological and economic issues.

PEO2 – TECHNICAL ACCOMPLISHMENTS

To provide knowledge based services to satisfy the needs of society and the industry by providing hands on experience in various technologies in core field.

PEO3 – INVENTION, INNOVATION AND CREATIVITY

To make the students to design, experiment, analyze, interpret in the core field with the help of other multi-disciplinary concepts wherever applicable.

PEO4 – PROFESSIONAL ETHICS

To educate the students to disseminate research findings with good soft skills and become a successful entrepreneur.

PEO5 – HUMAN RESOURCE DEVELOPMENT

To graduate the students in building national capabilities in technology, education and research.

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 understandingof 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 abilityto 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: R20A1283

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 datasets.

COURSE OUTCOMES:

1. Ability to add mining algorithms as a component to the existing tools
2. Demonstrate the classification, clustering and etc. in large data sets.
3. 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.	√	√	√								√

DATAWARE HOUSE TOOLS

Cloudera	cloudera
Teradata	TERADATA
Oracle	ORACLE
TabLeau	tableau SOFTWARE

OPEN SOURCE DATA MINING TOOLS

WEKA	WEKA The University of Waikato
Orange	orange DATA MINING FRUITFUL&FUN
KNIME	KNIME
R-Programming	R

DATA WAREHOUSING AND DATA MINING LAB

INDEX

Experiment 1: Installation of WEKA Tool

Aim: A. Investigation the Application interfaces of the Weka tool. Introduction:

Introduction

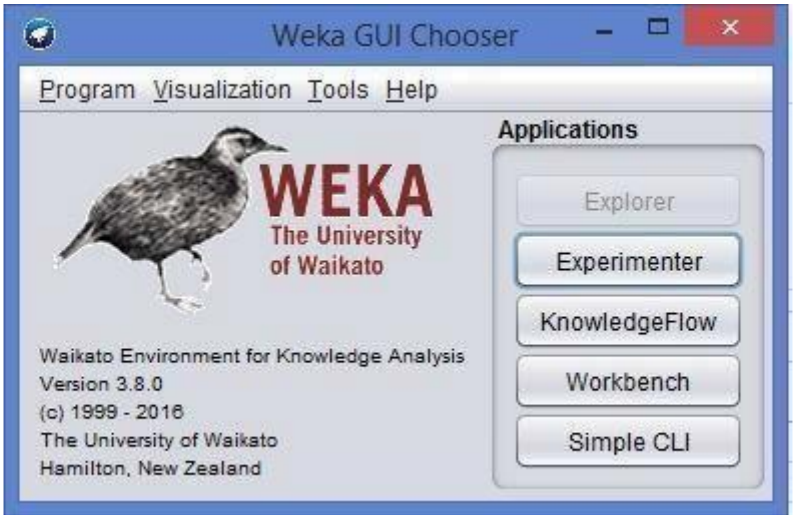
Weka (pronounced to rhyme with Mecca) is a workbench that contains a collection of visualization tools and algorithms for data analysis and predictive modeling, together with graphical user interfaces for easy access to these functions. The original non-Java version of Weka was a Tcl/Tk front-end to (mostly third-party) modeling algorithms implemented in other programming languages, plus data preprocessing utilities in C, and Make file-based system for running machine learning experiments. This original version was primarily designed as a tool for analyzing data from agricultural domains, but the more recent fully Java-based version (Weka 3), for which development started in 1997, is now used in many different application areas, in particular for educational purposes and research. Advantages of Weka include:

- Free availability under the GNU General Public License.
- Portability, since it is fully implemented in the Java programming language and thus runs on almost any modern computing platform
- A comprehensive collection of data preprocessing and modeling techniques
- Ease of use due to its graphical user interfaces

Description:

Open the program. Once the program has been loaded on the user’s machine it is opened by navigating to the programs start option and that will depend on the users operating system. Figure 1.1 is an example of the initial opening screen on a computer.

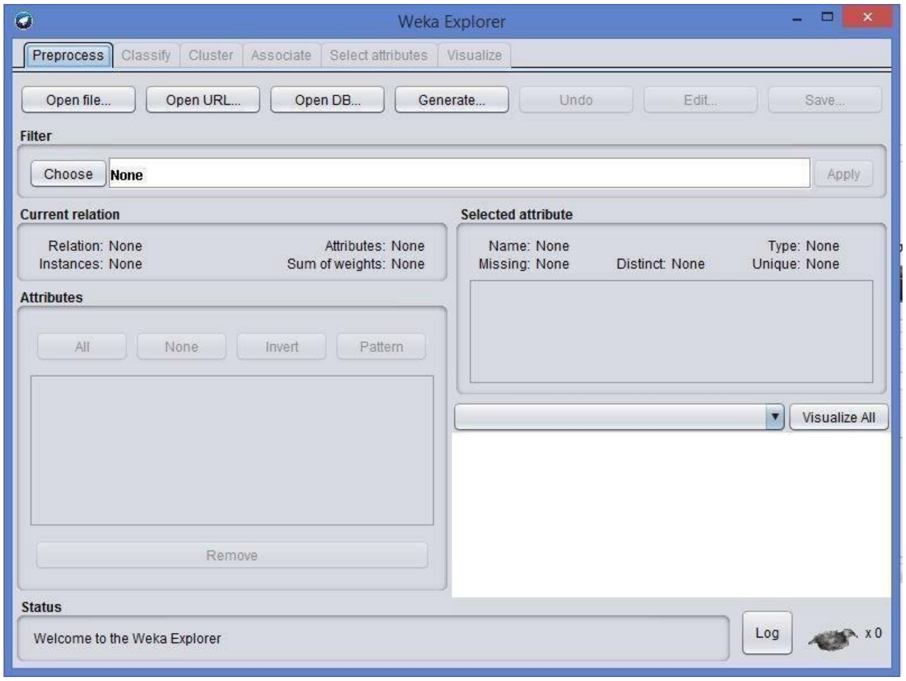
There are four options available on this initial screen:



Waikato Environment for Knowledge Analysis
Version 3.8.0
(c) 1999 - 2016
The University of Waikato
Hamilton, New Zealand

Fig: 1.1 Weka GUI

1. Explorer - the graphical interface used to conduct experimentation on raw data After clicking the Explorer button the weka explorer interface appears.



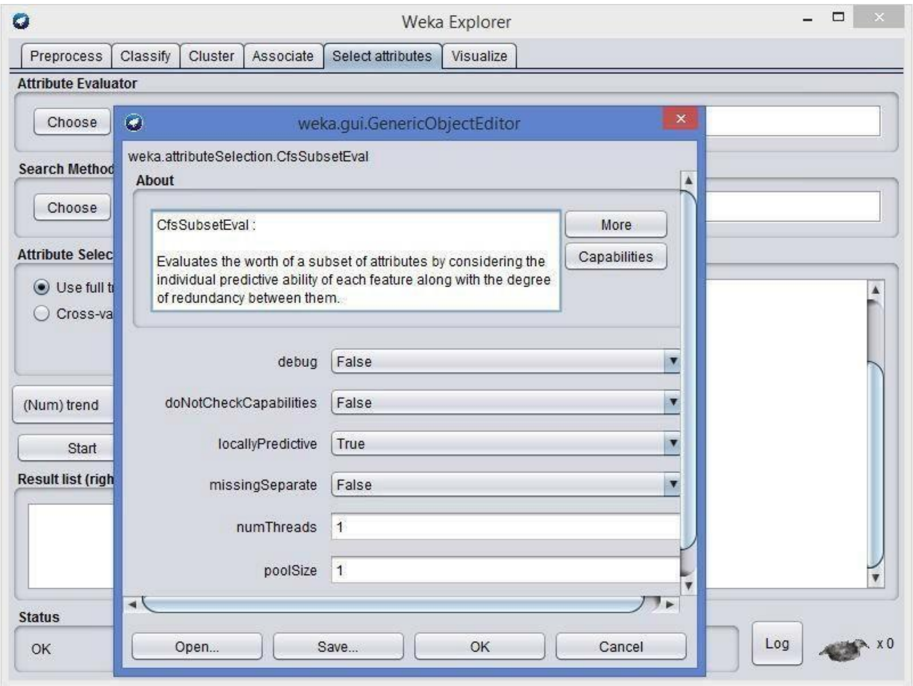
Filter: None

Relation: None
Instances: None
Attributes: None
Sum of weights: None

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

Welcome to the Weka Explorer

Fig: 1.2 Pre-processor



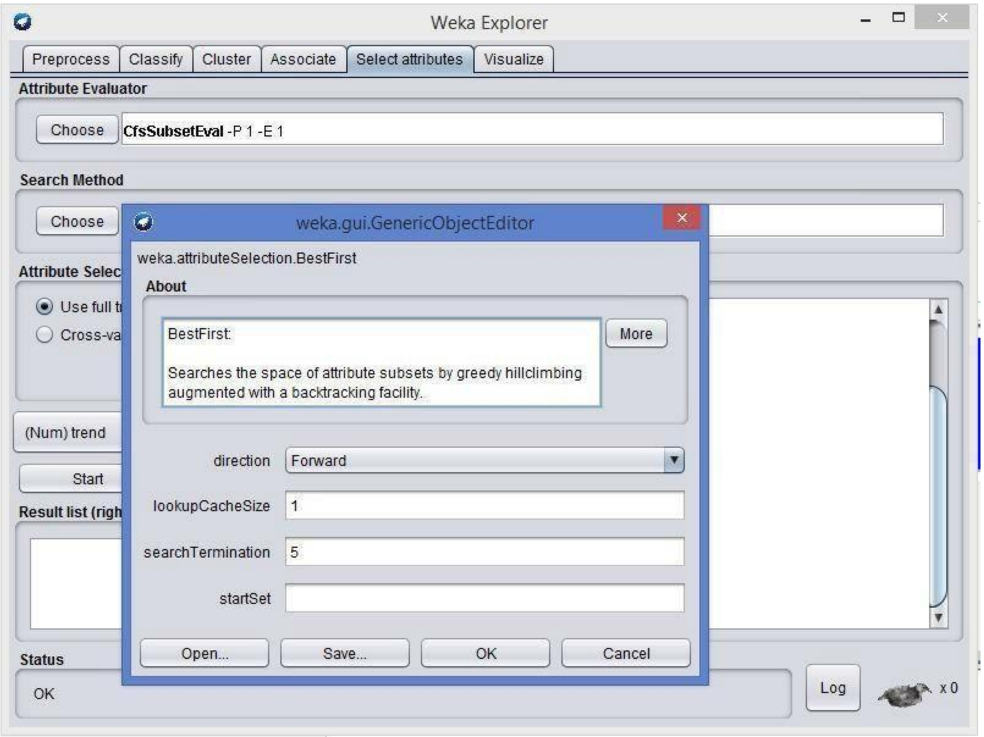
weka.attributeSelection.CfsSubsetEval

About

CfsSubsetEval :

Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them.

debug	False
doNotCheckCapabilities	False
locallyPredictive	True
missingSeparate	False
numThreads	1
poolSize	1



CfsSubsetEval -P 1 -E 1

weka.attributeSelection.BestFirst

About

BestFirst:

Searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility.

direction	Forward
lookupCacheSize	1
searchTermination	5
startSet	

Inside the weka explorer window there are six tabs:

1. **Preprocess-** used to choose the data file to be used by the application.

   Open File- allows for the user to select files residing on the local machine or recorded medium

   Open URL- provides a mechanism to locate a file or data source from a different location specified by the user

   Open Database- allows the user to retrieve files or data from a database source provided by user

2. **Classify-** used to test and train different learning schemes on the preprocessed data file under experimentation

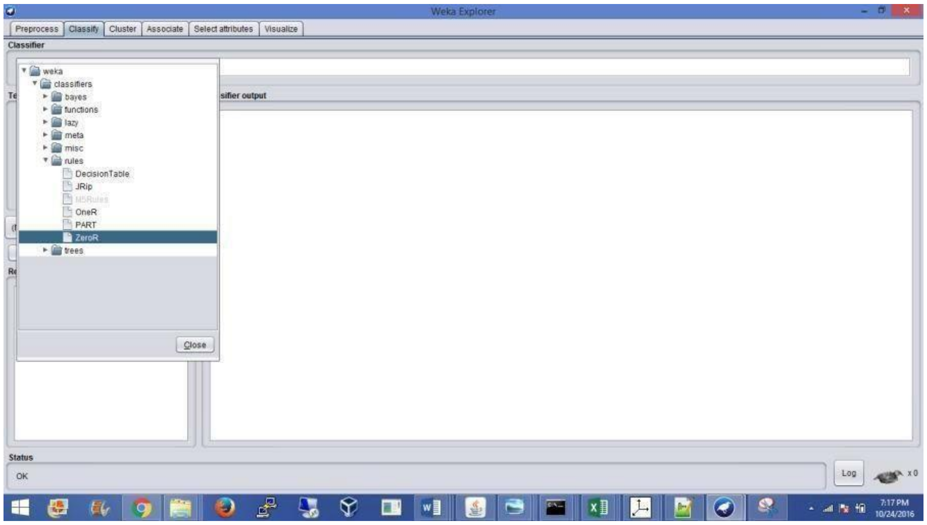


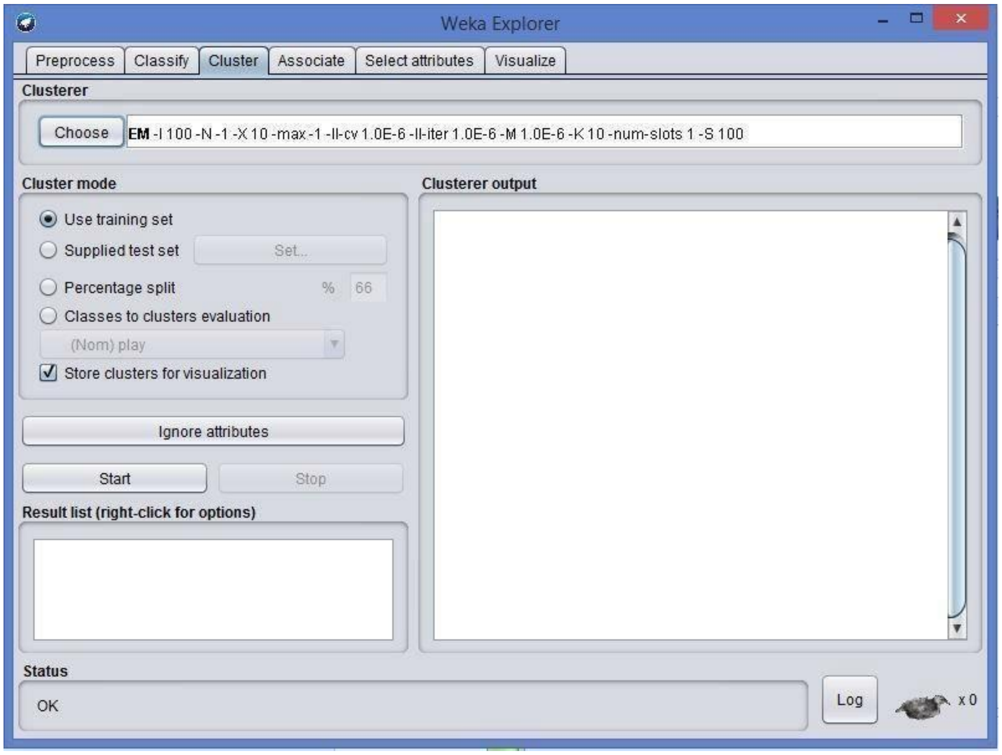
Fig: 1.3 choosing Zero set from classify

Again there are several options to be selected inside of the classify tab. Test option gives the user the choice of using four different test mode scenarios on the data set.

1. Use training set
2. Supplied training set
3. Cross validation
4. Split percentage

3. **Cluster-** used to apply different tools that identify clusters within the data file.

The Cluster tab opens the process that is used to identify commonalties or clusters of occurrences within the data set and produce information for the user to analyze.



Clusterer

EM -I 100 -N -1 -X 10 -max -1 -ll-cv 1.0E-6 -ll-iter 1.0E-6 -M 1.0E-6 -K 10 -num-slots 1 -S 100

Cluster mode

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

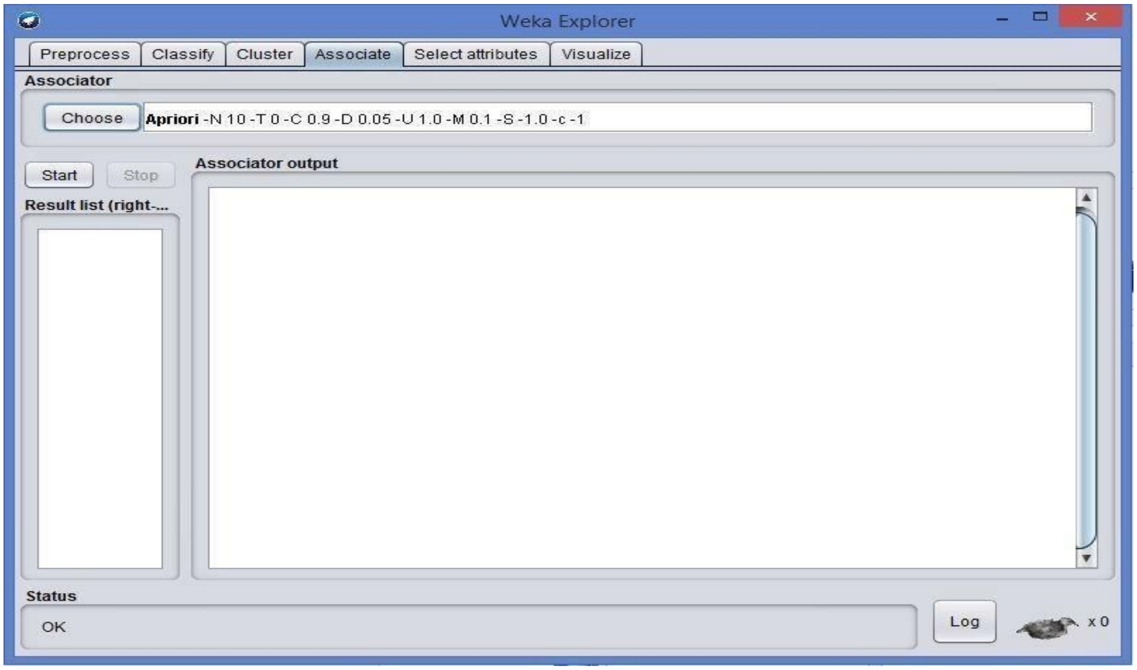
Clusterer output

Result list (right-click for options)

Status

OK

4. **Association-** used to apply different rules to the data file that identify association within the data. The associate tab opens a window to select the options for associations within thedataset.



Associator

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

Associator output

Result list (right-...

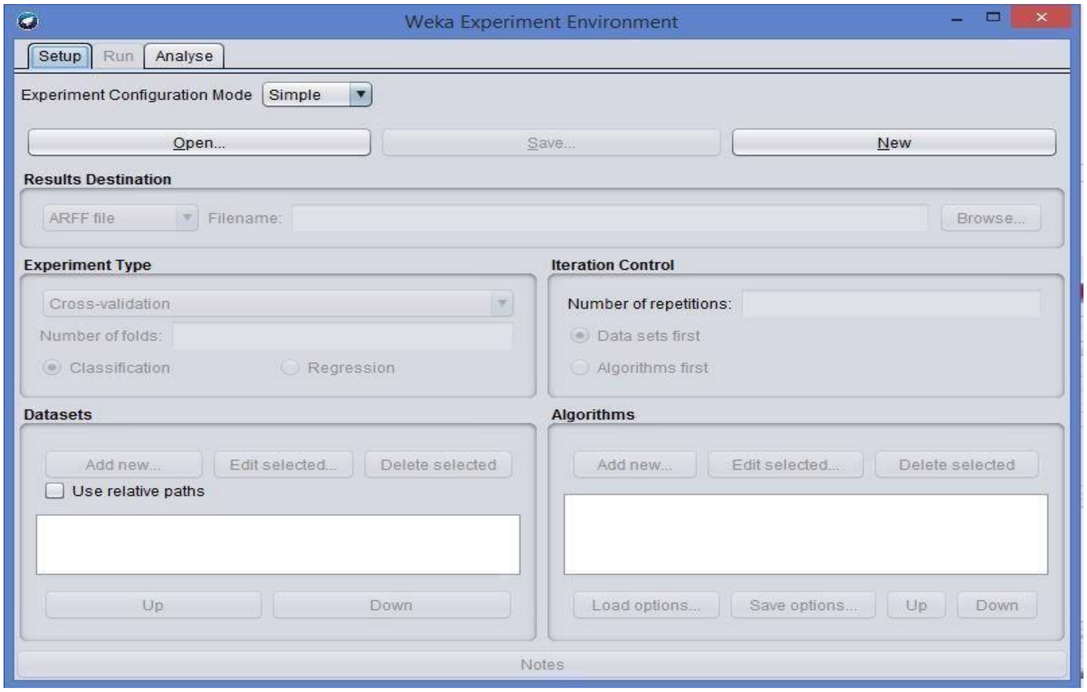
Status

OK

5. **Select attributes-**used to apply different rules to reveal changes based on selected attributes inclusion or exclusion from the experiment

6. **Visualize-** used to see what the various manipulation produced on the data set in a 2D format, in scatter plot and bar graph output.

2. Experimenter - this option allows users to conduct different experimental variations on data sets and perform statistical manipulation. 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.



Experiment Configuration Mode: Simple

ARFF file

Filename:

Cross-validation

Number of folds:

● Classification ○ Regression

Number of repetitions:

● Data sets first

○ Algorithms first

□ Use relative paths

Fig: 1.6 Weka experiment

Results destination: ARFF file, CSV file, JDBC database.

Experiment type: Cross-validation (default), Train/Test Percentage Split (data randomized).

Iteration control: Number of repetitions, Data sets first/Algorithms first.

Algorithms: filters

3. Knowledge Flow -basically the same functionality as Explorer with drag and drop functionality. The advantage of this option is that it supports incremental learning from previous results

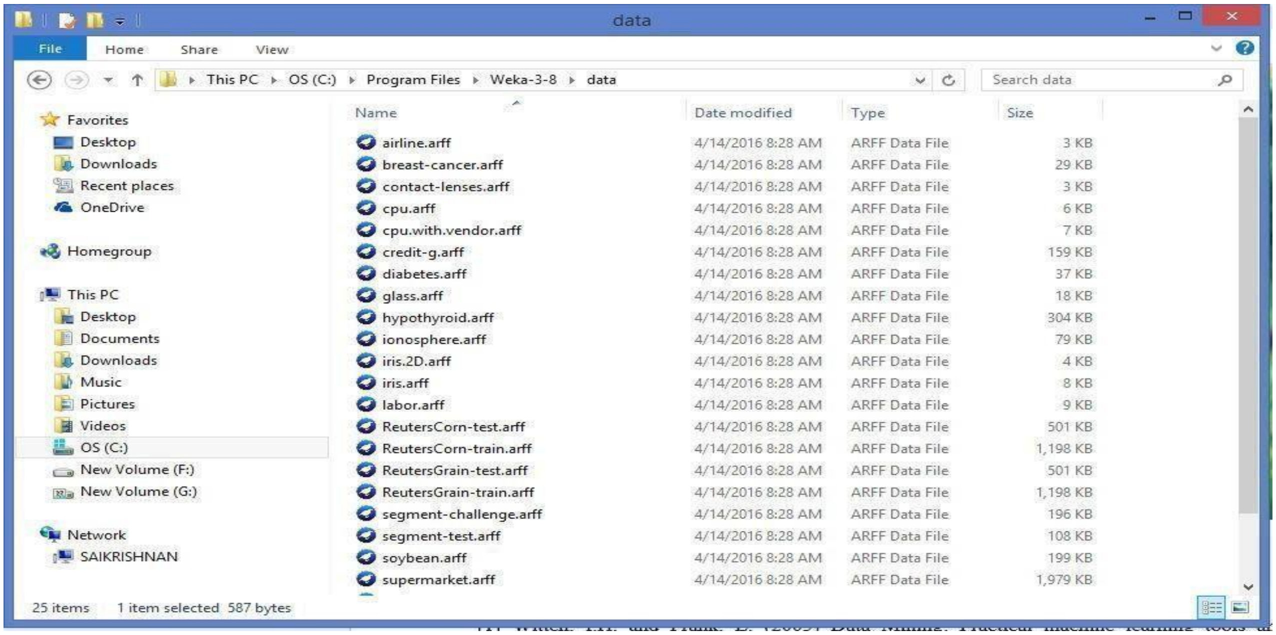
4. Simple CLI - provides users without a graphic interface option the ability to execute commands from a terminal window.

b. Explore the default datasets in weka tool.

Click the "***Open file…***" button to open a data set and double click on the "***data***" directory.

Weka provides a number of small common machine learning datasets that you can use to practice on.

Select the "***iris.arff***" file to load the Iris dataset.



Name	Date modified	Type	Size
airline.arff	4/14/2016 8:28 AM	ARFF Data File	3 KB
breast-cancer.arff	4/14/2016 8:28 AM	ARFF Data File	29 KB
contact-lenses.arff	4/14/2016 8:28 AM	ARFF Data File	3 KB
cpu.arff	4/14/2016 8:28 AM	ARFF Data File	6 KB
cpu.with.vendor.arff	4/14/2016 8:28 AM	ARFF Data File	7 KB
credit-g.arff	4/14/2016 8:28 AM	ARFF Data File	159 KB
diabetes.arff	4/14/2016 8:28 AM	ARFF Data File	37 KB
glass.arff	4/14/2016 8:28 AM	ARFF Data File	18 KB
hypothyroid.arff	4/14/2016 8:28 AM	ARFF Data File	304 KB
ionosphere.arff	4/14/2016 8:28 AM	ARFF Data File	79 KB
iris.2D.arff	4/14/2016 8:28 AM	ARFF Data File	4 KB
iris.arff	4/14/2016 8:28 AM	ARFF Data File	8 KB
labor.arff	4/14/2016 8:28 AM	ARFF Data File	9 KB
ReutersCorn-test.arff	4/14/2016 8:28 AM	ARFF Data File	501 KB
ReutersCorn-train.arff	4/14/2016 8:28 AM	ARFF Data File	1,198 KB
ReutersGrain-test.arff	4/14/2016 8:28 AM	ARFF Data File	501 KB
ReutersGrain-train.arff	4/14/2016 8:28 AM	ARFF Data File	1,198 KB
segment-challenge.arff	4/14/2016 8:28 AM	ARFF Data File	196 KB
segment-test.arff	4/14/2016 8:28 AM	ARFF Data File	108 KB
soybean.arff	4/14/2016 8:28 AM	ARFF Data File	199 KB
supermarket.arff	4/14/2016 8:28 AM	ARFF Data File	1,979 KB

Fig: 1.7 Different Data Sets in weka

References:

[1] Witten, I.H. and Frank, E. (2005) Data Mining: Practical machine learning tools andtechniques. 2nd edition Morgan Kaufmann, San Francisco.
[2] Ross Quinlan (1993). C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA.
[3] CVS–http://weka.sourceforge.net/wiki/index.php/CVS
[4] Weka Doc–http://weka.sourceforge.net/wekadoc/

Exercise:

1. Normalize the data using min-max normalization

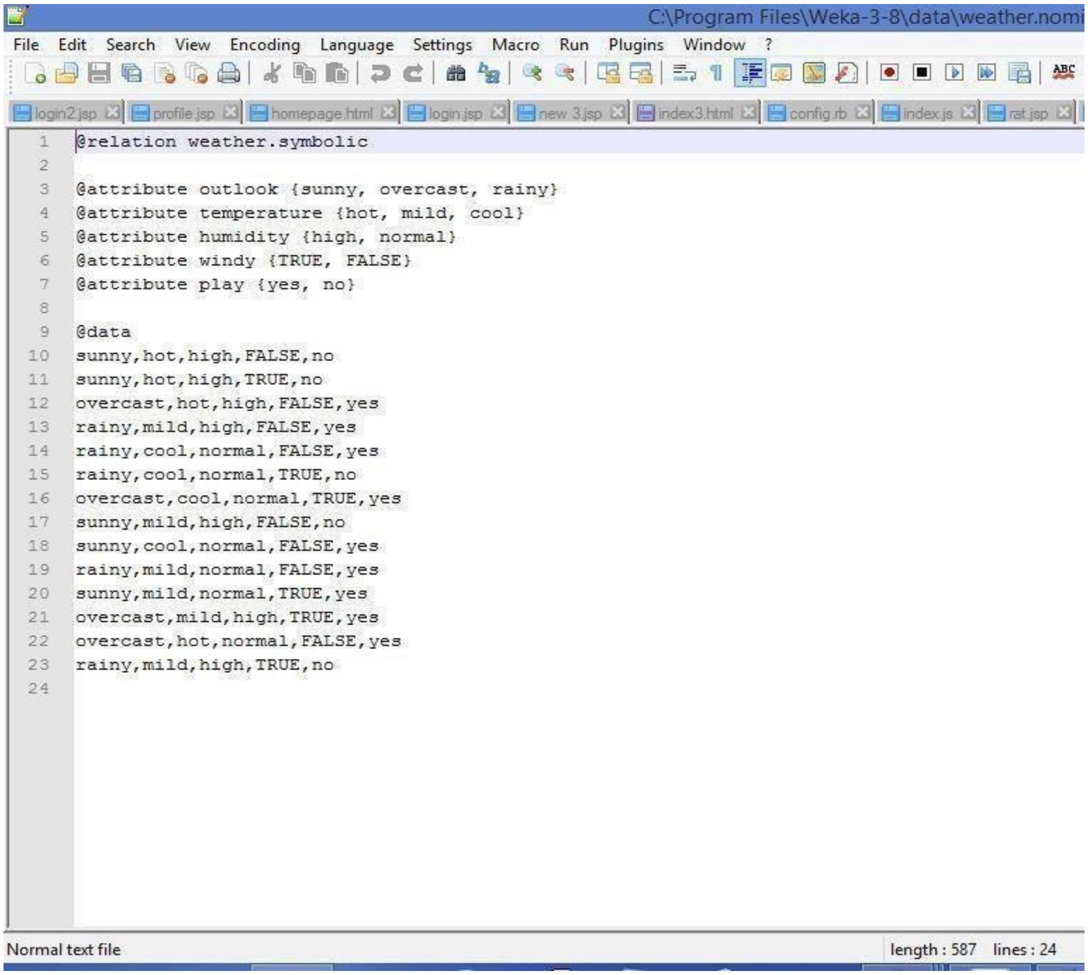
Record Notes

Experiment 2: Creating new ARFF file

Aim: Creating a new ARFF file

An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes. ARFF files were developed by the Machine Learning Project at the Department of Computer Science of The University of Waikato for use with the Weka machine learning software in WEKA, each data entry is an instance of the java class weka.core. Instance, and each instance consists of a For loading datasets in WEKA, WEKA can load ARFF files. Attribute Relation File Format has two sections:

1. The Header section defines relation (dataset) name, attribute name, and type.
2. The Data section lists the data instances.



```
@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
```

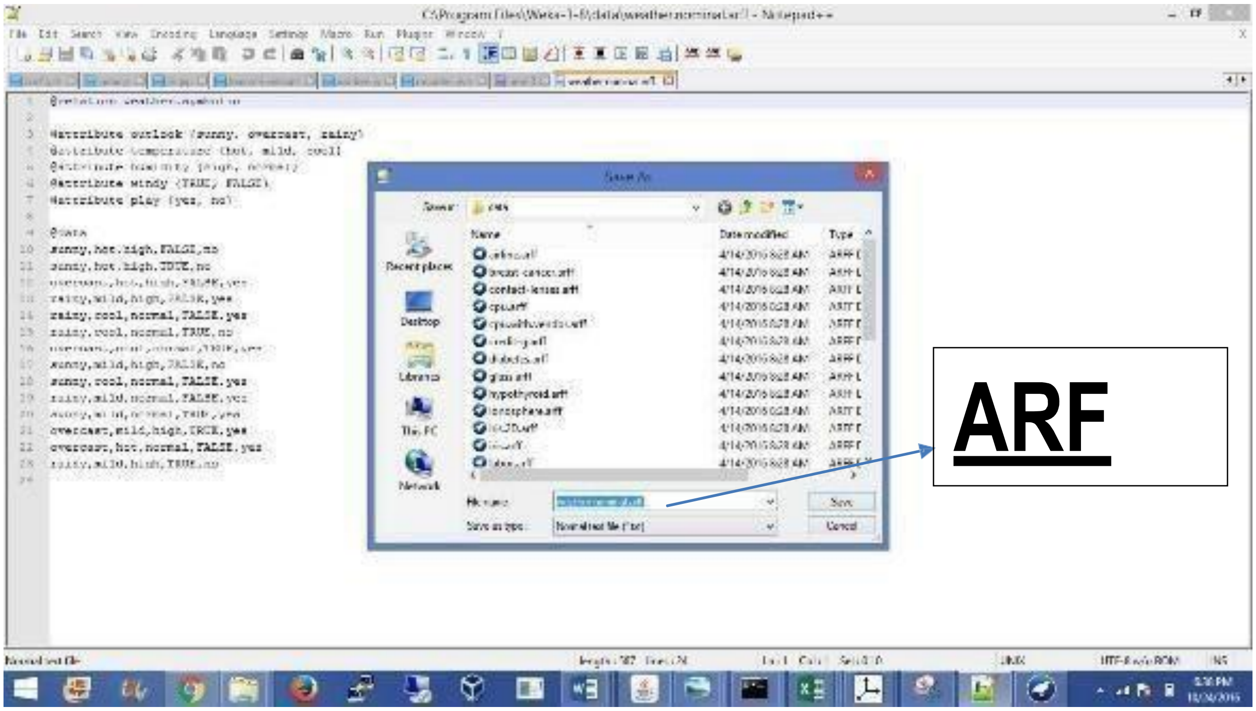
The figure above is from the textbook that shows an ARFF file for the weather data. Lines beginning with a % sign are comments. And there are three basic keywords:

 "@relation" in Header section, followed with relation name.

 "@attribute" in Header section, followed with attributes name and its type (or range).
 "@data" in Data section, followed with the list of data instances.

The external representation of an Instances class Consists of:

- **A header:** Describes the attribute types
- **Data section:** Comma separated list of data



```
@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
```

References:

https://www.cs.auckland.ac.nz/courses/compsci367s1c/tutorials/IntroductionToWeka.pdf

Exercise:

1. Creating a sample dataset for supermarket (supermarket.arff)

Record Notes

Experiment 3: Data Processing Techniques on Data Set

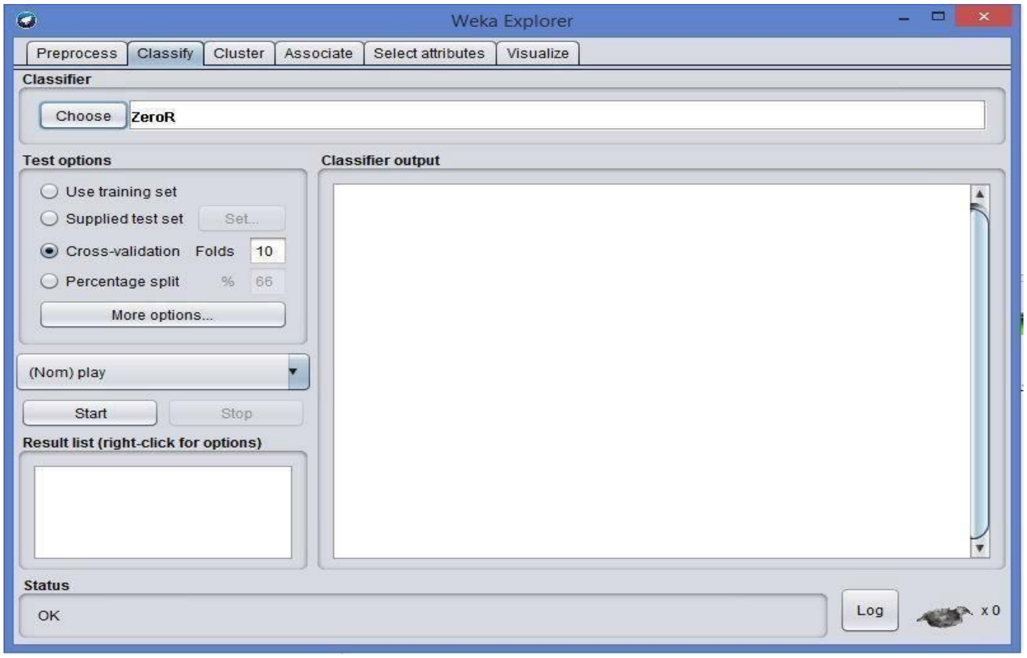
Aim: 3a) Pre-process a given dataset based on Attribute selection

To search through all possible combinations of attributes in the data and find which subset of attributes works best for prediction, make sure that you set up attribute evaluator to „Cfs Subset Val‟ and a search method to „Best First‟. The evaluator will determine what method to use toassign a worth to each subset of attributes. The search method will determine what style of search to perform. The options that you can set for selection in the „Attribute Selection Mode‟ fig no: 3.2

1. **Use full training set.** The worth of the attribute subset is determined using the full set of training data.

2. **Cross-validation.** The worth of the attribute subset is determined by a process of cross-validation. The „Fold‟ and „Seed‟ fields set the number of folds to use and the random seed used when shuffling the data.

Specify which attribute to treat as the class in the drop-down box below the test options. Once all the test options are set, you can start the attribute selection process by clicking on „Start‟ button.



Weka Explorer

Classifier: ZeroR

Test options

○ Use training set

○ Supplied test set

● Cross-validation Folds 10

○ Percentage split % 66

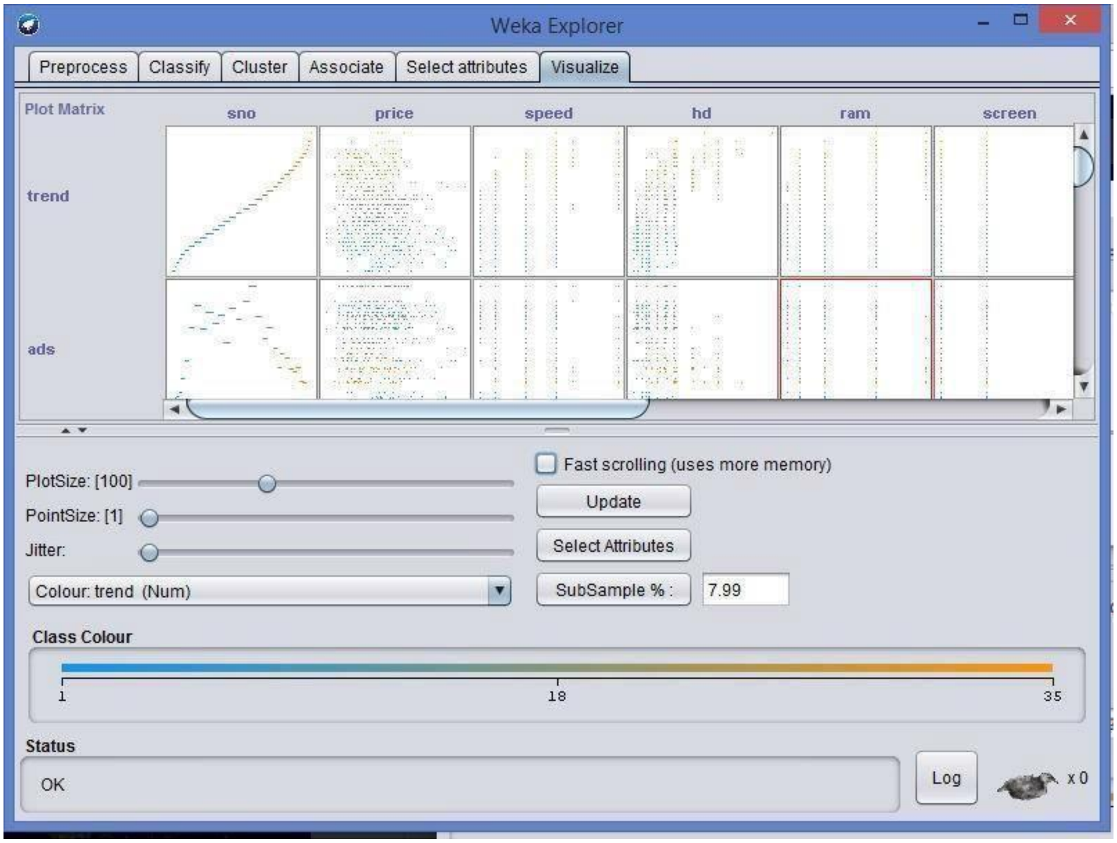
(Nom) play

Status: OK

Fig: 3.1 Choosing Cross validation

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

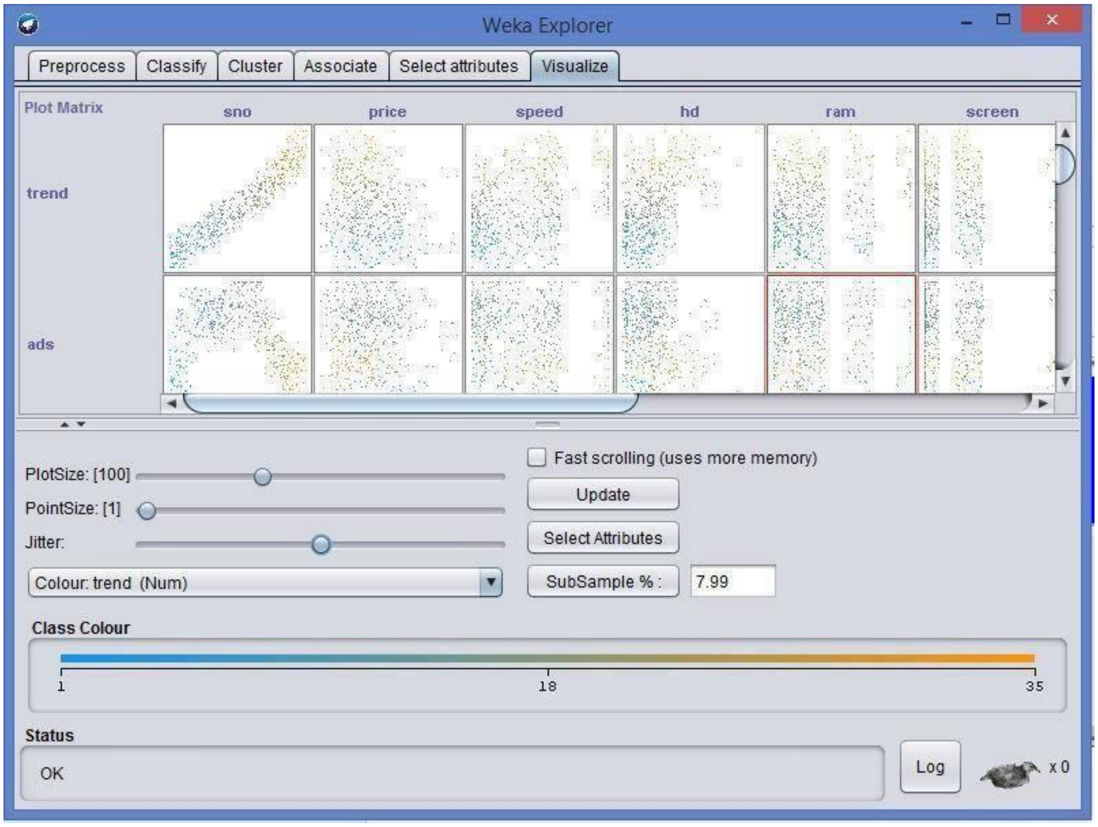
2. Visualizing Results



PlotSize: [100]
PointSize: [1]
Jitter:
Colour: trend (Num)
SubSample % : 7.99
Status
OK

Fig: 3.2 Data Visualization

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



PlotSize: [100]

PointSize: [1]

Jitter:

Colour: trend (Num)

□ Fast scrolling (uses more memory)

SubSample % : 7.99

Status

OK

Fig 3.3: Preprocessing with jitter

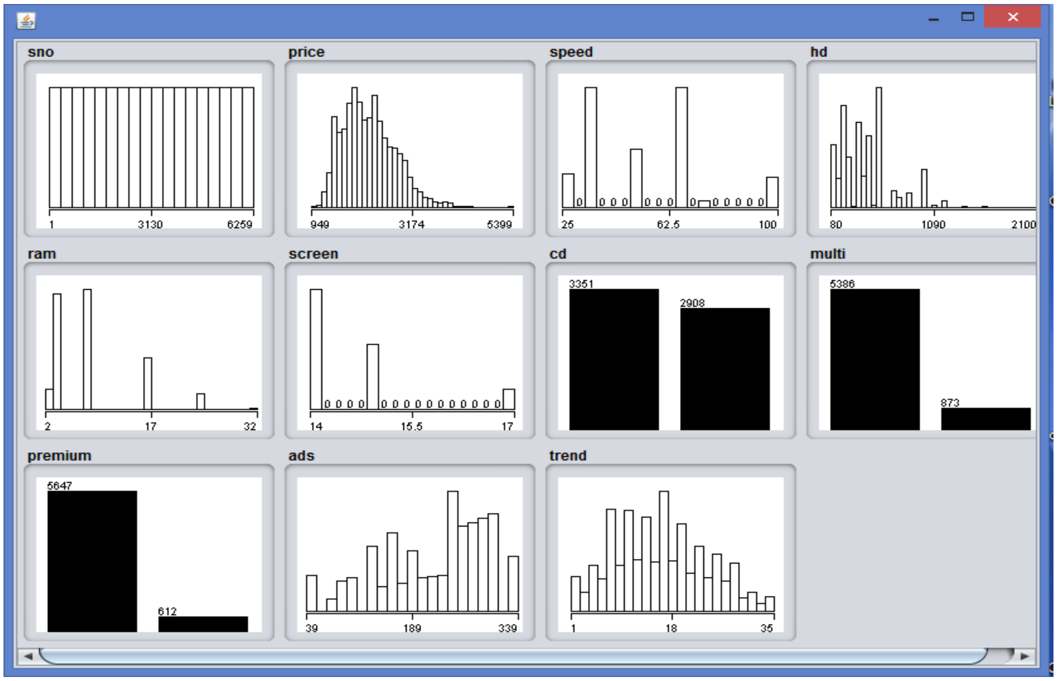


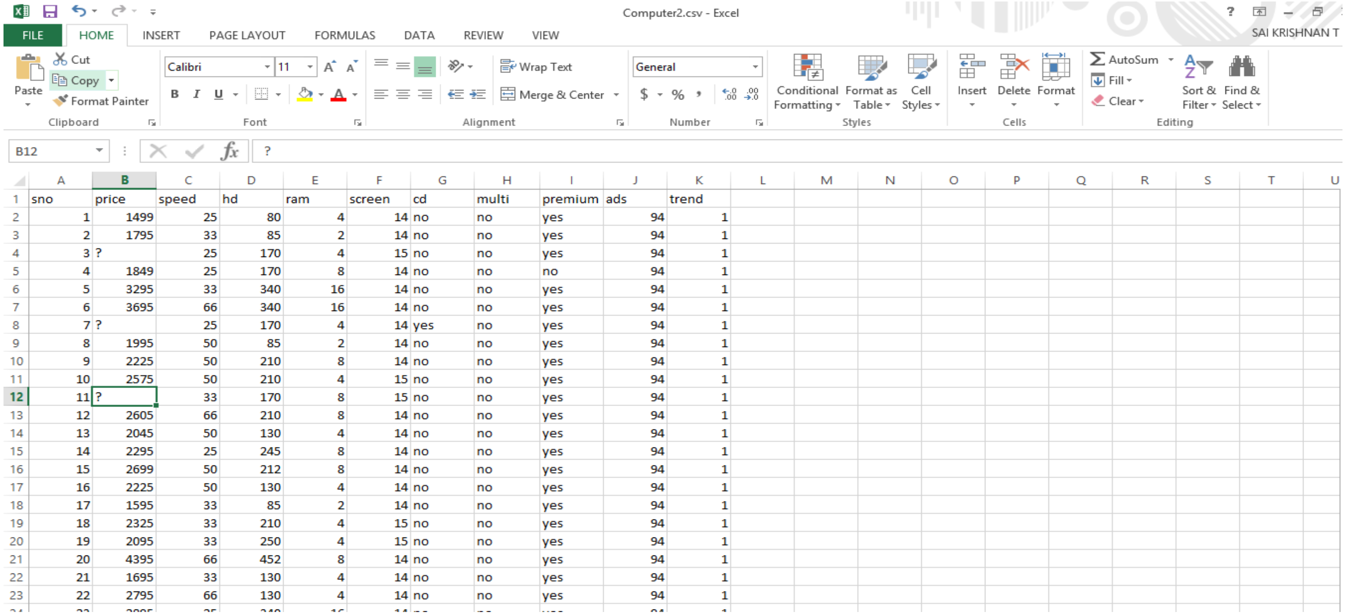
Fig: 3.3 Data visualization

Exercise

1. Explain data preprocessing steps for heart disease dataset.

Aim: B. Pre-process a given dataset based on Handling Missing Values

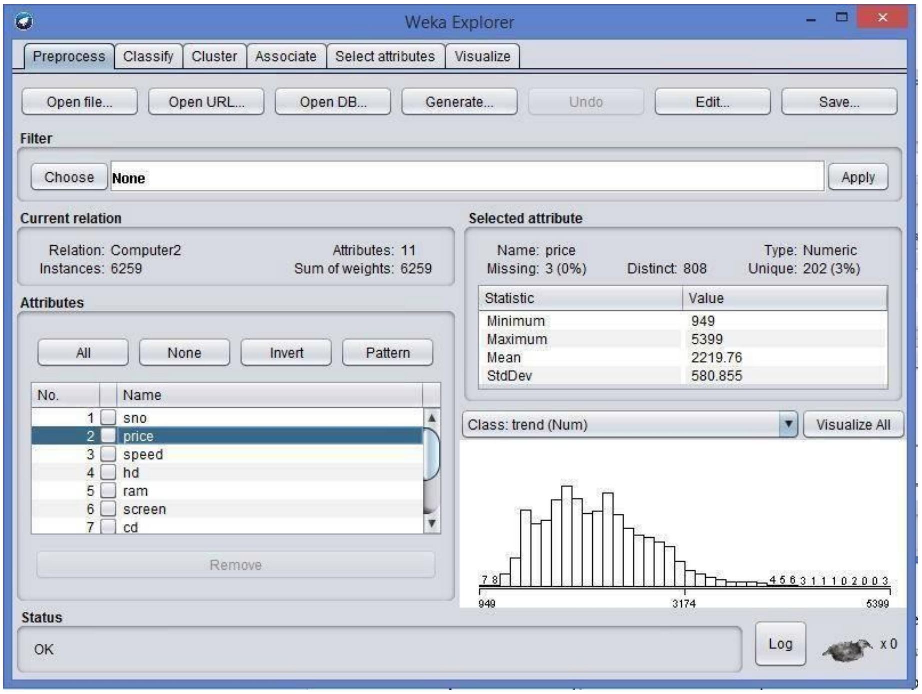
Process: Replacing Missing Attribute Values by the Attribute Mean. This method is used for data sets with numerical attributes. An example of such a data set is presented in fig no: 3.4



B12 ?

	A	B	C	D	E	F	G	H	I	J	K
1	sno	price	speed	hd	ram	screen	cd	multi	premium	ads	trend
2	1	1499	25	80	4	14	no	no	yes	94	1
3	2	1795	33	85	2	14	no	no	yes	94	1
4	3	?	25	170	4	15	no	no	yes	94	1
5	4	1849	25	170	8	14	no	no	no	94	1
6	5	3295	33	340	16	14	no	no	yes	94	1
7	6	3695	66	340	16	14	no	no	yes	94	1
8	7	?	25	170	4	14	yes	no	yes	94	1
9	8	1995	50	85	2	14	no	no	yes	94	1
10	9	2225	50	210	8	14	no	no	yes	94	1
11	10	2575	50	210	4	15	no	no	yes	94	1
12	11	?	33	170	8	15	no	no	yes	94	1
13	12	2605	66	210	8	14	no	no	yes	94	1
14	13	2045	50	130	4	14	no	no	yes	94	1
15	14	2295	25	245	8	14	no	no	yes	94	1
16	15	2699	50	212	8	14	no	no	yes	94	1
17	16	2225	50	130	4	14	no	no	yes	94	1
18	17	1595	33	85	2	14	no	no	yes	94	1
19	18	2325	33	210	4	15	no	no	yes	94	1
20	19	2095	33	250	4	15	no	no	yes	94	1
21	20	4395	66	452	8	14	no	no	yes	94	1
22	21	1695	33	130	4	14	no	no	yes	94	1
23	22	2795	66	130	4	14	no	no	yes	94	1

Fig: 3.4 Missing values



Filter: None

Current relation
Relation: Computer2
Instances: 6259
Attributes: 11
Sum of weights: 6259

Selected attribute
Name: price
Missing: 3 (0%)
Distinct: 808
Type: Numeric
Unique: 202 (3%)

Statistic	Value
Minimum	949
Maximum	5399
Mean	2219.76
StdDev	580.855

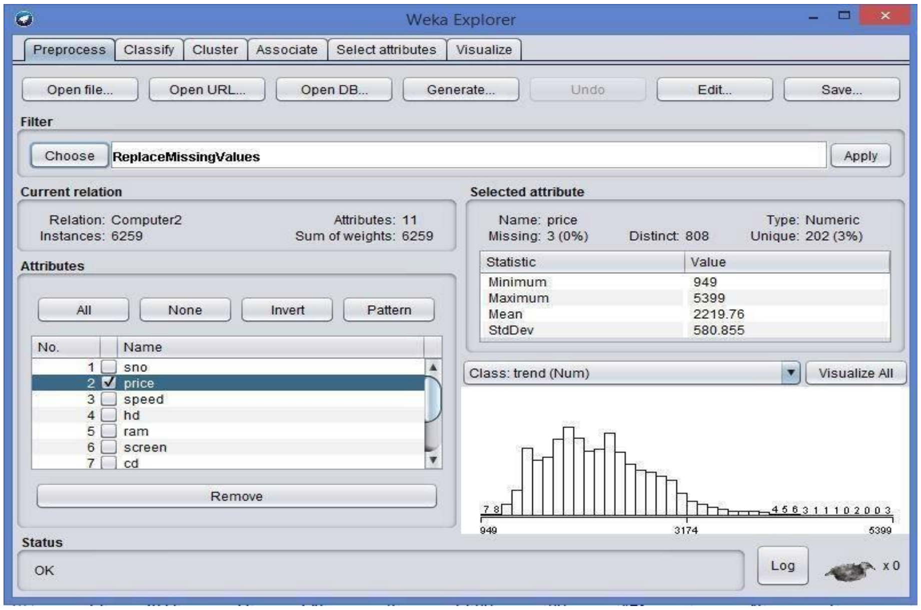
Attributes

No.	Name
1	☐ sno
2	☐ price
3	☐ speed
4	☐ hd
5	☐ ram
6	☐ screen
7	☐ cd

Status
OK

Fig: 3.5 Choosing a dataset

In this method, every missing attribute value for a numerical attribute is replaced by the arithmetic mean of known attribute values. In Fig, the mean of known attribute values for Temperature is 99.2, hence all missing attribute values for Temperature should be replaced by The table with missing attribute values replaced by the mean is presented in fig. For symbolic attributes Headache and Nausea, missing attribute values were replaced using the most common value of the Replace Missing Values.



Filter: ReplaceMissingValues

Current relation
Relation: Computer2 Attributes: 11
Instances: 6259 Sum of weights: 6259

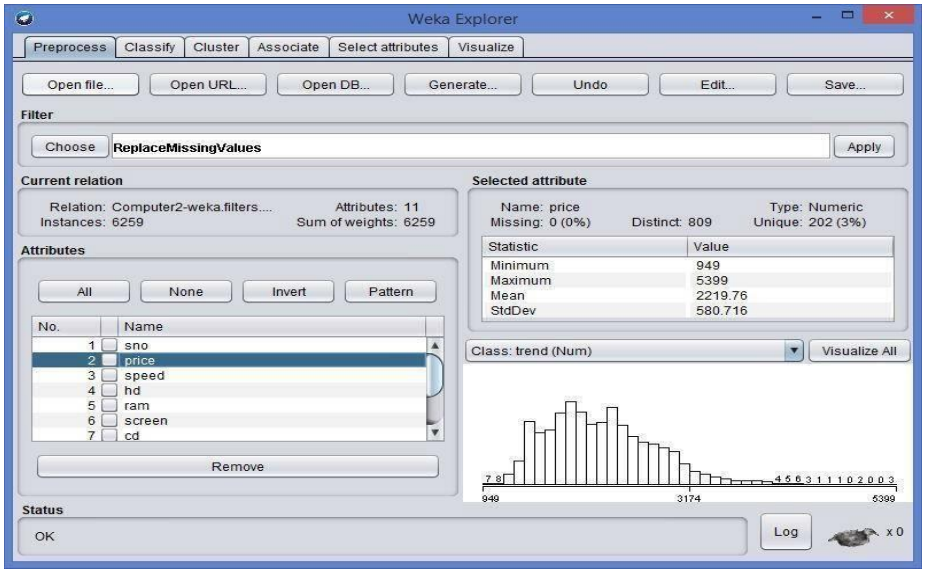
Selected attribute
Name: price Type: Numeric
Missing: 3 (0%) Distinct: 808 Unique: 202 (3%)

Statistic	Value
Minimum	949
Maximum	5399
Mean	2219.76
StdDev	580.855

No.		Name
1	□	sno
2	☑	price
3	□	speed
4	□	hd
5	□	ram
6	□	screen
7	□	cd

Class: trend (Num)

Status
OK



Filter: ReplaceMissingValues

Current relation
Relation: Computer2-weka.filters.... Attributes: 11
Instances: 6259 Sum of weights: 6259

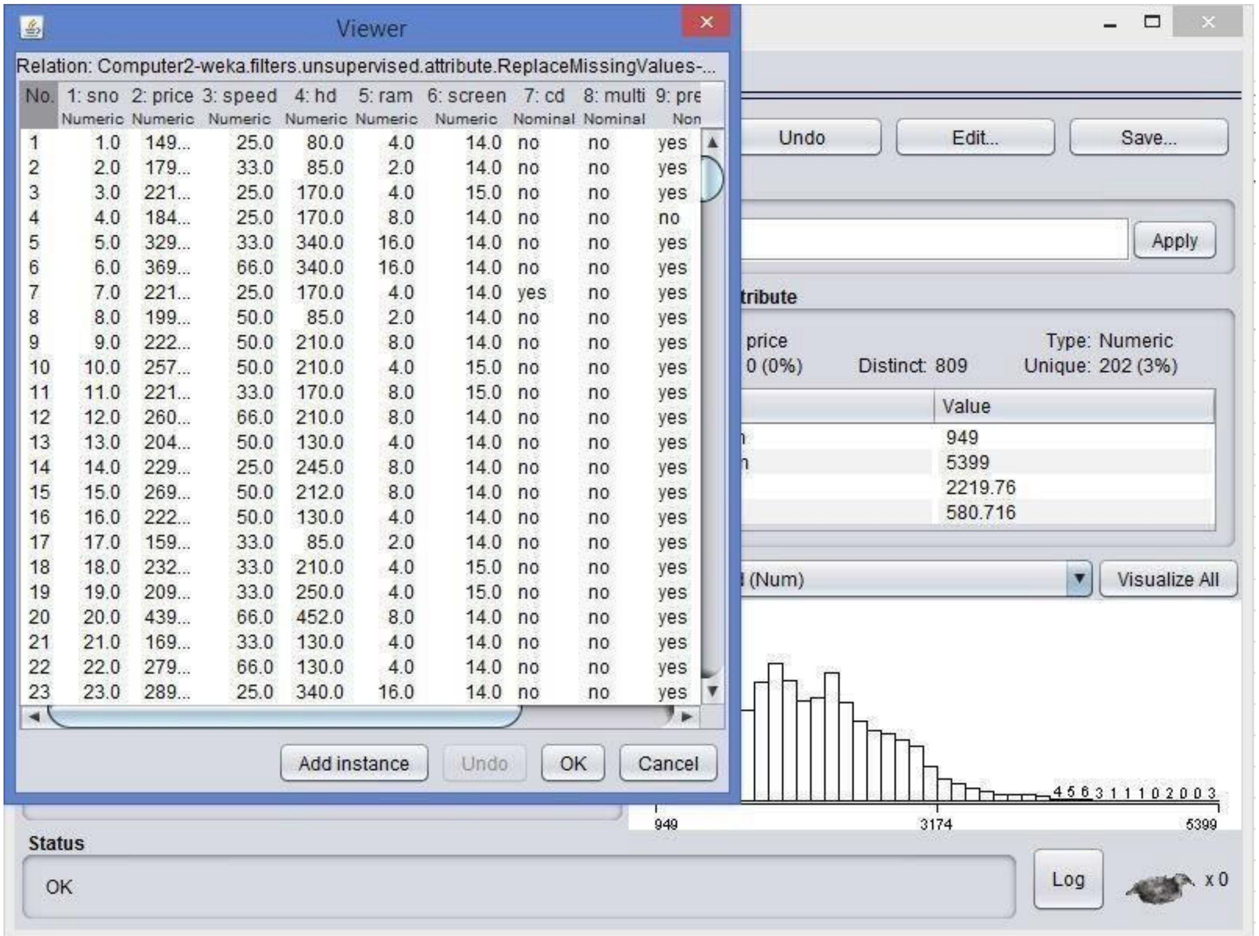
Selected attribute
Name: price Type: Numeric
Missing: 0 (0%) Distinct: 809 Unique: 202 (3%)

Statistic	Value
Minimum	949
Maximum	5399
Mean	2219.76
StdDev	580.716

No.		Name
1	□	sno
2	□	price
3	□	speed
4	□	hd
5	□	ram
6	□	screen
7	□	cd

Class: trend (Num)

Status
OK



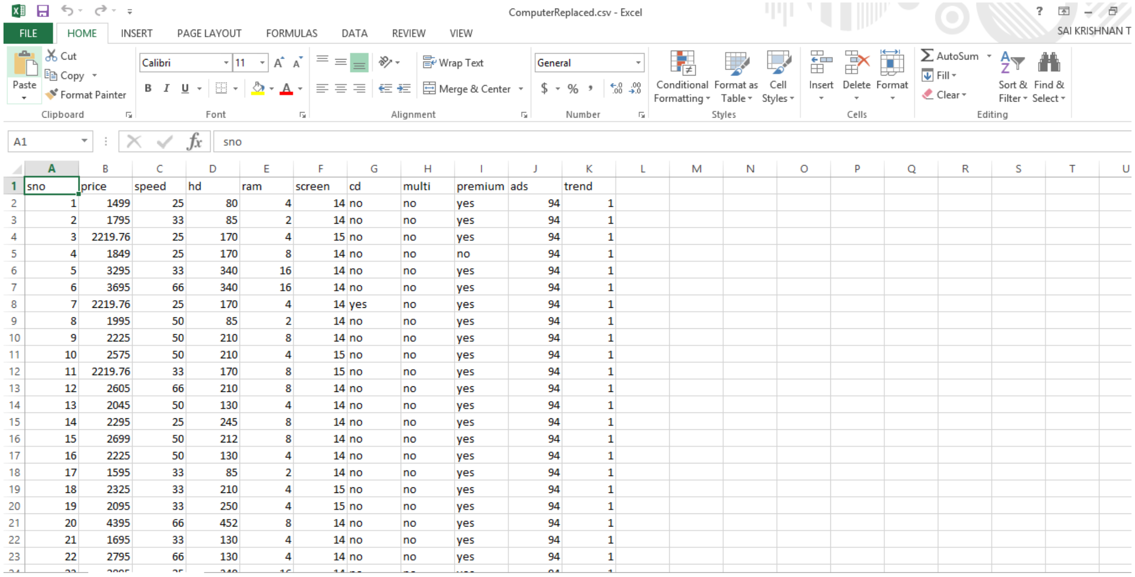
Relation: Computer2-weka.filters.unsupervised.attribute.ReplaceMissingValues-...

No.	1: sno Numeric	2: price Numeric	3: speed Numeric	4: hd Numeric	5: ram Numeric	6: screen Numeric	7: cd Nominal	8: multi Nominal	9: pre Non
1	1.0	149...	25.0	80.0	4.0	14.0	no	no	yes
2	2.0	179...	33.0	85.0	2.0	14.0	no	no	yes
3	3.0	221...	25.0	170.0	4.0	15.0	no	no	yes
4	4.0	184...	25.0	170.0	8.0	14.0	no	no	no
5	5.0	329...	33.0	340.0	16.0	14.0	no	no	yes
6	6.0	369...	66.0	340.0	16.0	14.0	no	no	yes
7	7.0	221...	25.0	170.0	4.0	14.0	yes	no	yes
8	8.0	199...	50.0	85.0	2.0	14.0	no	no	yes
9	9.0	222...	50.0	210.0	8.0	14.0	no	no	yes
10	10.0	257...	50.0	210.0	4.0	15.0	no	no	yes
11	11.0	221...	33.0	170.0	8.0	15.0	no	no	yes
12	12.0	260...	66.0	210.0	8.0	14.0	no	no	yes
13	13.0	204...	50.0	130.0	4.0	14.0	no	no	yes
14	14.0	229...	25.0	245.0	8.0	14.0	no	no	yes
15	15.0	269...	50.0	212.0	8.0	14.0	no	no	yes
16	16.0	222...	50.0	130.0	4.0	14.0	no	no	yes
17	17.0	159...	33.0	85.0	2.0	14.0	no	no	yes
18	18.0	232...	33.0	210.0	4.0	15.0	no	no	yes
19	19.0	209...	33.0	250.0	4.0	15.0	no	no	yes
20	20.0	439...	66.0	452.0	8.0	14.0	no	no	yes
21	21.0	169...	33.0	130.0	4.0	14.0	no	no	yes
22	22.0	279...	66.0	130.0	4.0	14.0	no	no	yes
23	23.0	289...	25.0	340.0	16.0	14.0	no	no	yes

price
0 (0%)
Distinct: 809
Type: Numeric
Unique: 202 (3%)

	Value
	949
	5399
	2219.76
	580.716

Status
OK



	A	B	C	D	E	F	G	H	I	J	K
1	sno	price	speed	hd	ram	screen	cd	multi	premium	ads	trend
2	1	1499	25	80	4	14	no	no	yes	94	1
3	2	1795	33	85	2	14	no	no	yes	94	1
4	3	2219.76	25	170	4	15	no	no	yes	94	1
5	4	1849	25	170	8	14	no	no	no	94	1
6	5	3295	33	340	16	14	no	no	yes	94	1
7	6	3695	66	340	16	14	no	no	yes	94	1
8	7	2219.76	25	170	4	14	yes	no	yes	94	1
9	8	1995	50	85	2	14	no	no	yes	94	1
10	9	2225	50	210	8	14	no	no	yes	94	1
11	10	2575	50	210	4	15	no	no	yes	94	1
12	11	2219.76	33	170	8	15	no	no	yes	94	1
13	12	2605	66	210	8	14	no	no	yes	94	1
14	13	2045	50	130	4	14	no	no	yes	94	1
15	14	2295	25	245	8	14	no	no	yes	94	1
16	15	2699	50	212	8	14	no	no	yes	94	1
17	16	2225	50	130	4	14	no	no	yes	94	1
18	17	1595	33	85	2	14	no	no	yes	94	1
19	18	2325	33	210	4	15	no	no	yes	94	1
20	19	2095	33	250	4	15	no	no	yes	94	1
21	20	4395	66	452	8	14	no	no	yes	94	1
22	21	1695	33	130	4	14	no	no	yes	94	1
23	22	2795	66	130	4	14	no	no	yes	94	1

Fig: 3.6 Replaced values

Exercise

1. Create your own dataset having missing values included.

Record Notes

Experiment 4: Data cube construction – OLAP operations

An OLAP cube is a term that typically refers to multi-dimensional array of data. OLAP is an acronym for online analytical processing,[1]which is a computer-based technique of analyzing data to look for insights. The term cube here refers to a multi-dimensional dataset, which is also sometimes called a hypercube if the number of dimensions is greater than 3.

Operations:

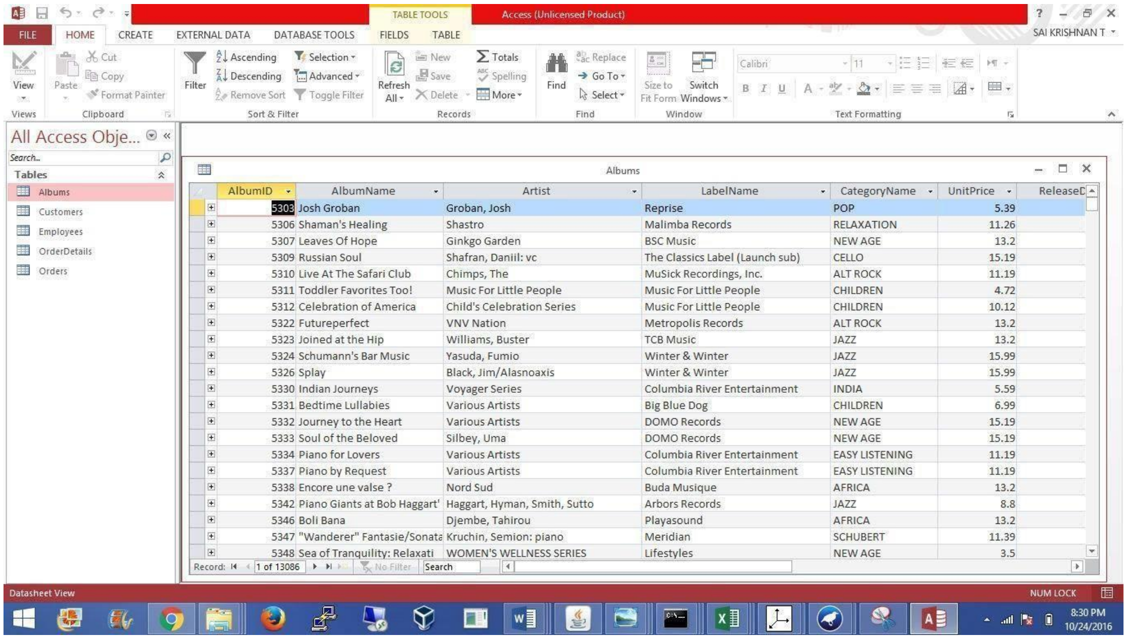
1. **Slice** is the act of picking a rectangular subset of a cube by choosing a single value for one of its dimensions, creating a new cube with one fewer dimension.[4] The picture shows a slicing operation: The sales figures of all sales regions and all product categories of the company in the year 2005 and 2006 are "sliced" out of the data cube.

2. **Dice:** The dice operation produces a subcube by allowing the analyst to pick specific values of multiple dimensions.[5]The picture shows a dicing operation: The new cube shows the sales figures of a limited number of product categories, the time and region dimensions cover the same range as before.

3. **Drill Down/Up** allows the user to navigate among levels of data ranging from the most summarized (up) to the most detailed (down).[4] The picture shows a drill-down operation: The analyst moves from the summary category "Outdoor-Schutzausrüstung" to see the sales figures for the individual products.

4. **Roll-up:** A roll-up involves summarizing the data along a dimension. The summarization rule might be computing totals along a hierarchy or applying a set of formulas such as "profit = sales - expenses".

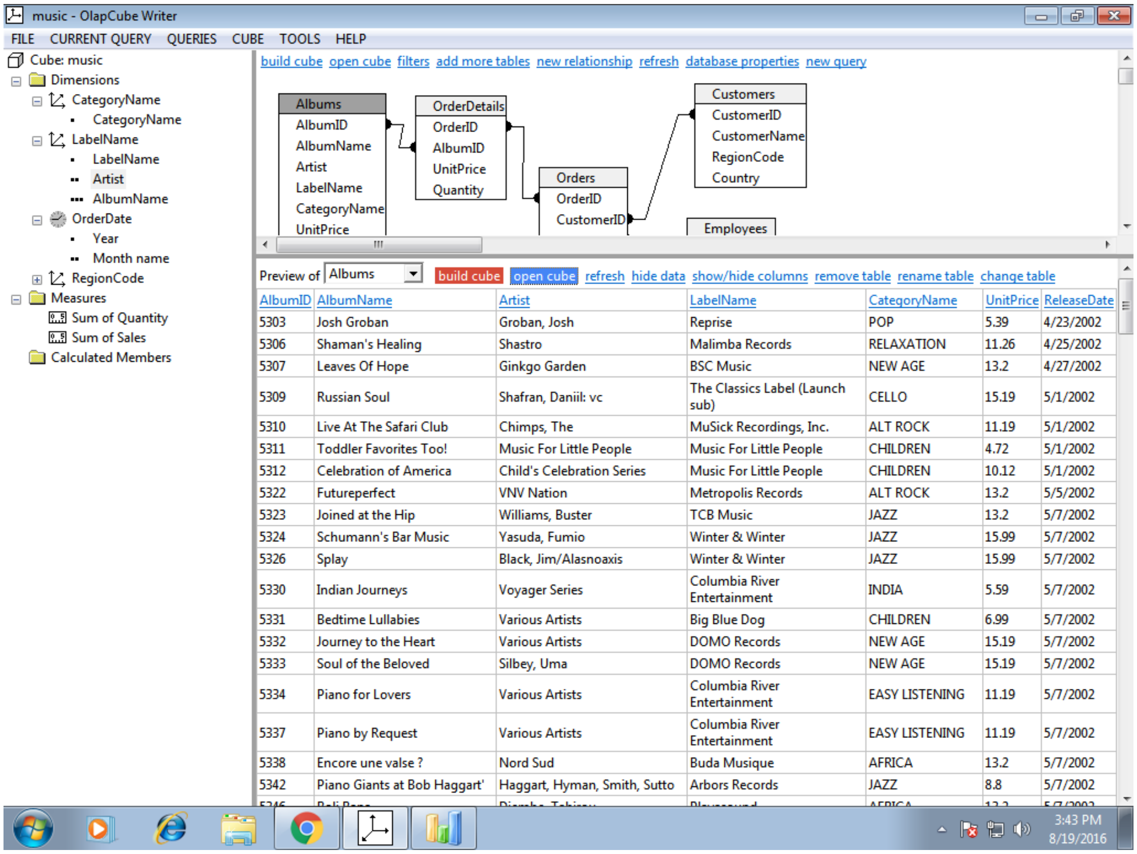
5. **Pivot** allows an analyst to rotate the cube in space to see its various faces. For example, cities could be arranged vertically and products horizontally while viewing data for a particular quarter. Pivoting could replace products with time periods to see data across time for a single product.



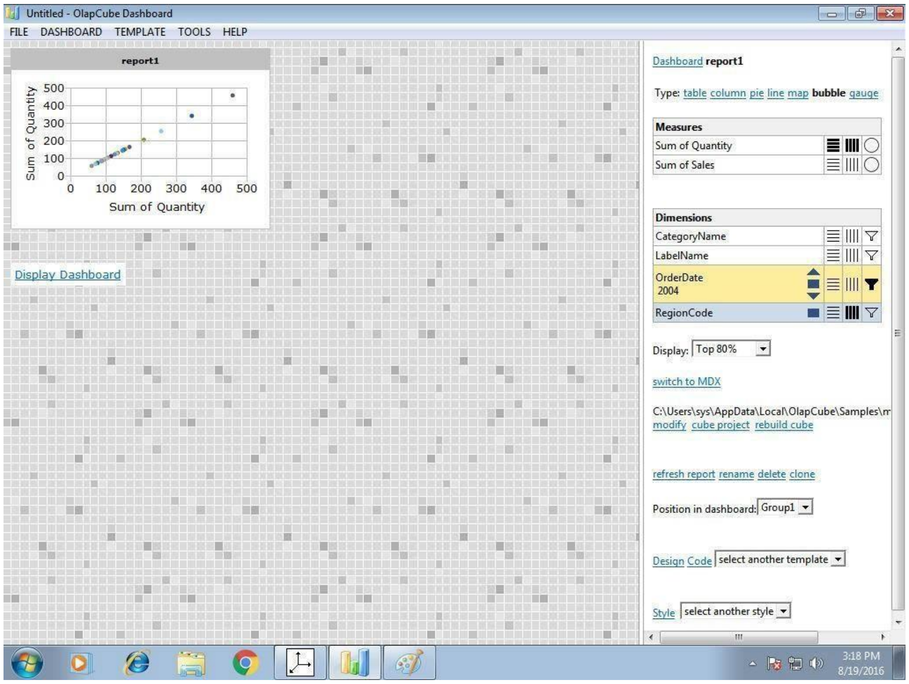
Albums

AlbumID	AlbumName	Artist	LabelName	CategoryName	UnitPrice	ReleaseD
5303	Josh Groban	Groban, Josh	Reprise	POP	5.39	
5306	Shaman's Healing	Shastro	Malimba Records	RELAXATION	11.26	
5307	Leaves Of Hope	Ginkgo Garden	BSC Music	NEW AGE	13.2	
5309	Russian Soul	Shafran, Daniil: vc	The Classics Label (Launch sub)	CELLO	15.19	
5310	Live At The Safari Club	Chimps, The	MuSick Recordings, Inc.	ALT ROCK	11.19	
5311	Toddler Favorites Too!	Music For Little People	Music For Little People	CHILDREN	4.72	
5312	Celebration of America	Child's Celebration Series	Music For Little People	CHILDREN	10.12	
5322	Futureperfect	VNV Nation	Metropolis Records	ALT ROCK	13.2	
5323	Joined at the Hip	Williams, Buster	TCB Music	JAZZ	13.2	
5324	Schumann's Bar Music	Yasuda, Fumio	Winter & Winter	JAZZ	15.99	
5326	Splay	Black, Jim/Alasnoaxis	Winter & Winter	JAZZ	15.99	
5330	Indian Journeys	Voyager Series	Columbia River Entertainment	INDIA	5.59	
5331	Bedtime Lullabies	Various Artists	Big Blue Dog	CHILDREN	6.99	
5332	Journey to the Heart	Various Artists	DOMO Records	NEW AGE	15.19	
5333	Soul of the Beloved	Silbey, Uma	DOMO Records	NEW AGE	15.19	
5334	Piano for Lovers	Various Artists	Columbia River Entertainment	EASY LISTENING	11.19	
5337	Piano by Request	Various Artists	Columbia River Entertainment	EASY LISTENING	11.19	
5338	Encore une valse ?	Nord Sud	Buda Musique	AFRICA	13.2	
5342	Piano Giants at Bob Haggart'	Haggart, Hyman, Smith, Sutto	Arbors Records	JAZZ	8.8	
5346	Boli Bana	Djembe, Tahirou	Playasound	AFRICA	13.2	
5347	"Wanderer" Fantasie/Sonata	Kruchin, Semion: piano	Meridian	SCHUBERT	11.39	
5348	Sea of Tranquility: Relaxati	WOMEN'S WELLNESS SERIES	Lifestyles	NEW AGE	3.5	

Record: 1 of 13086 No Filter Search



AlbumID	AlbumName	Artist	LabelName	CategoryName	UnitPrice	ReleaseDate
5303	Josh Groban	Groban, Josh	Reprise	POP	5.39	4/23/2002
5306	Shaman's Healing	Shastro	Malimba Records	RELAXATION	11.26	4/25/2002
5307	Leaves Of Hope	Ginkgo Garden	BSC Music	NEW AGE	13.2	4/27/2002
5309	Russian Soul	Shafran, Daniil: vc	The Classics Label (Launch sub)	CELLO	15.19	5/1/2002
5310	Live At The Safari Club	Chimps, The	MuSick Recordings, Inc.	ALT ROCK	11.19	5/1/2002
5311	Toddler Favorites Too!	Music For Little People	Music For Little People	CHILDREN	4.72	5/1/2002
5312	Celebration of America	Child's Celebration Series	Music For Little People	CHILDREN	10.12	5/1/2002
5322	Futureperfect	VNV Nation	Metropolis Records	ALT ROCK	13.2	5/5/2002
5323	Joined at the Hip	Williams, Buster	TCB Music	JAZZ	13.2	5/7/2002
5324	Schumann's Bar Music	Yasuda, Fumio	Winter & Winter	JAZZ	15.99	5/7/2002
5326	Splay	Black, Jim/Alasnoaxis	Winter & Winter	JAZZ	15.99	5/7/2002
5330	Indian Journeys	Voyager Series	Columbia River Entertainment	INDIA	5.59	5/7/2002
5331	Bedtime Lullabies	Various Artists	Big Blue Dog	CHILDREN	6.99	5/7/2002
5332	Journey to the Heart	Various Artists	DOMO Records	NEW AGE	15.19	5/7/2002
5333	Soul of the Beloved	Silbey, Uma	DOMO Records	NEW AGE	15.19	5/7/2002
5334	Piano for Lovers	Various Artists	Columbia River Entertainment	EASY LISTENING	11.19	5/7/2002
5337	Piano by Request	Various Artists	Columbia River Entertainment	EASY LISTENING	11.19	5/7/2002
5338	Encore une valse ?	Nord Sud	Buda Musique	AFRICA	13.2	5/7/2002
5342	Piano Giants at Bob Haggart'	Haggart, Hyman, Smith, Sutto	Arbors Records	JAZZ	8.8	5/7/2002
5346	Boli Bana	Djembe, Tahirou	Playasound	AFRICA	13.2	5/7/2002



Dashboard **report1**

Type: table column pie line map **bubble** gauge

Measures
Sum of Quantity
Sum of Sales

Dimensions
CategoryName
LabelName
OrderDate 2004
RegionCode

Display: Top 80%

switch to MDX

C:\Users\sys\AppData\Local\OlapCube\Samples\m

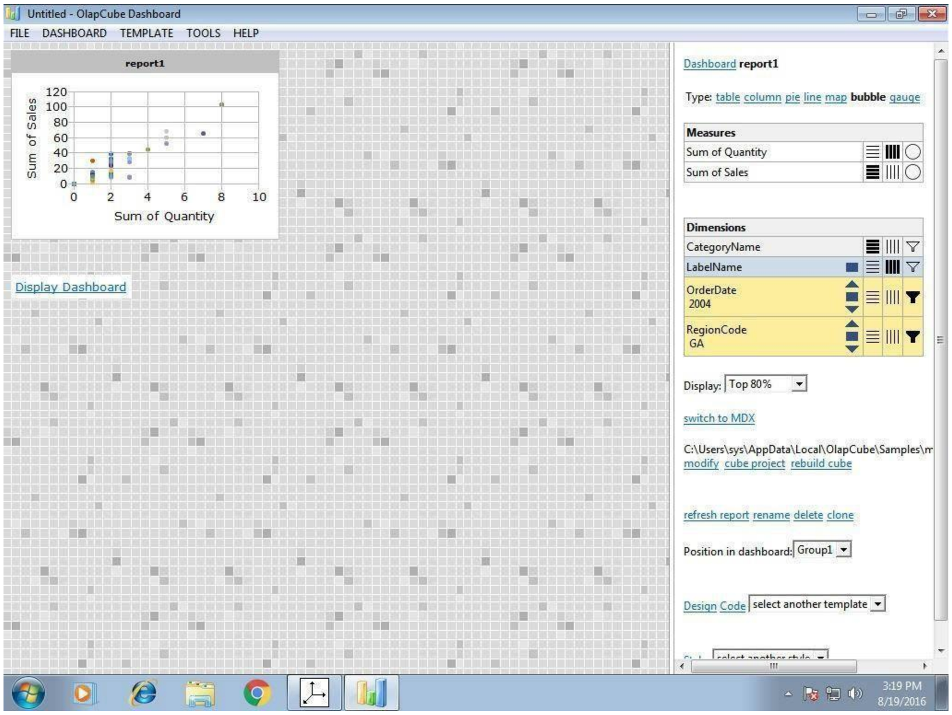
modify cube project rebuild cube

refresh report rename delete clone

Position in dashboard: Group1

Design Code select another template

Style select another style



Dashboard **report1**

Type: table column pie line map **bubble** gauge

Measures
Sum of Quantity
Sum of Sales

Dimensions
CategoryName
LabelName
OrderDate 2004
RegionCode GA

Display: Top 80%

switch to MDX

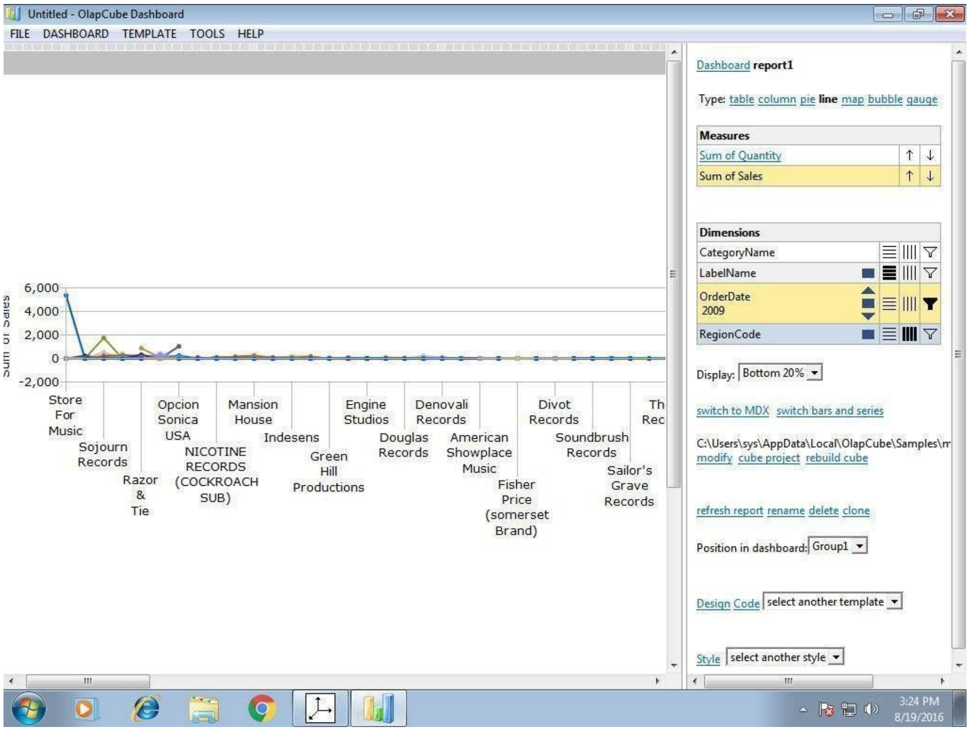
C:\Users\sys\AppData\Local\OlapCube\Samples\m

modify cube project rebuild cube

refresh report rename delete clone

Position in dashboard: Group1

Design Code select another template



Dashboard report1

Type: table column pie **line** map bubble gauge

Measures		
Sum of Quantity	↑	↓
Sum of Sales	↑	↓

Dimensions
CategoryName
LabelName
OrderDate 2009
RegionCode

Display: Bottom 20%

switch to MDX switch bars and series

C:\Users\sys\AppData\Local\OlapCube\Samples\m

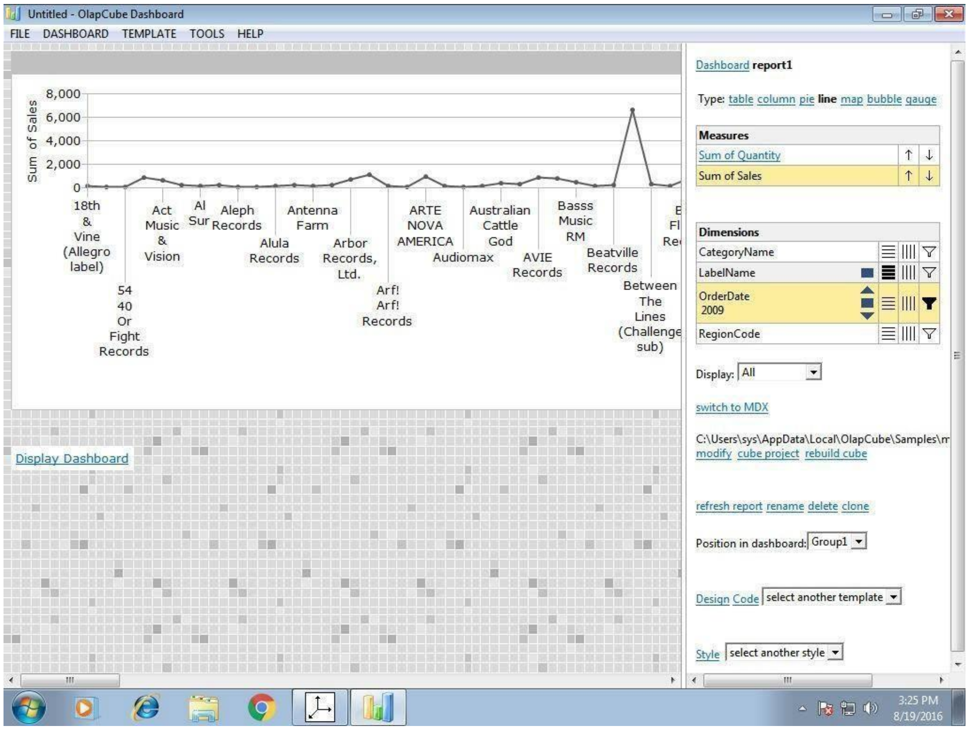
modify cube project rebuild cube

refresh report rename delete clone

Position in dashboard: Group1

Design Code select another template

Style select another style



Dashboard report1

Type: table column pie **line** map bubble gauge

Measures		
Sum of Quantity	↑	↓
Sum of Sales	↑	↓

Dimensions
CategoryName
LabelName
OrderDate 2009
RegionCode

Display: All

switch to MDX

C:\Users\sys\AppData\Local\OlapCube\Samples\m

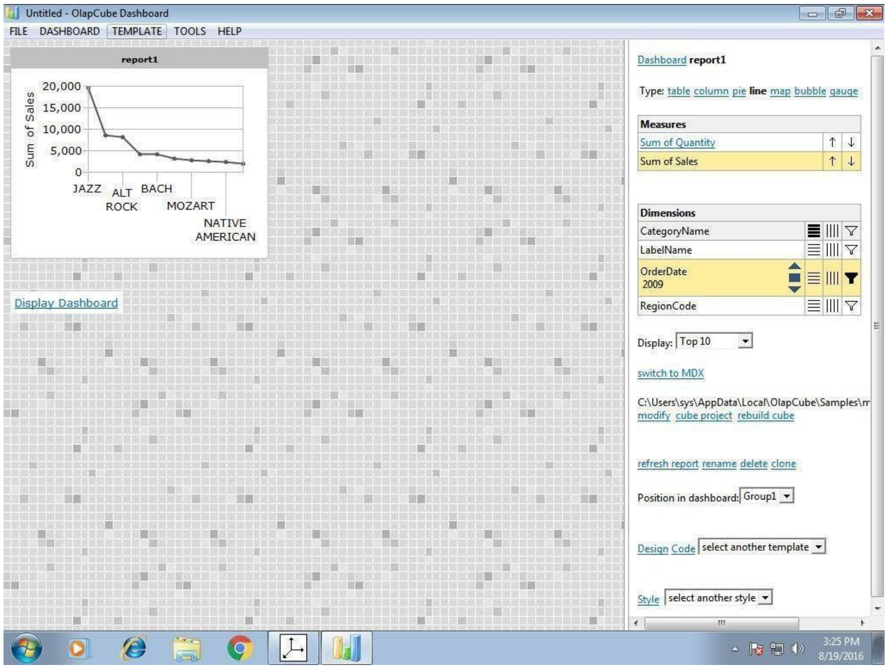
modify cube project rebuild cube

refresh report rename delete clone

Position in dashboard: Group1

Design Code select another template

Style select another style



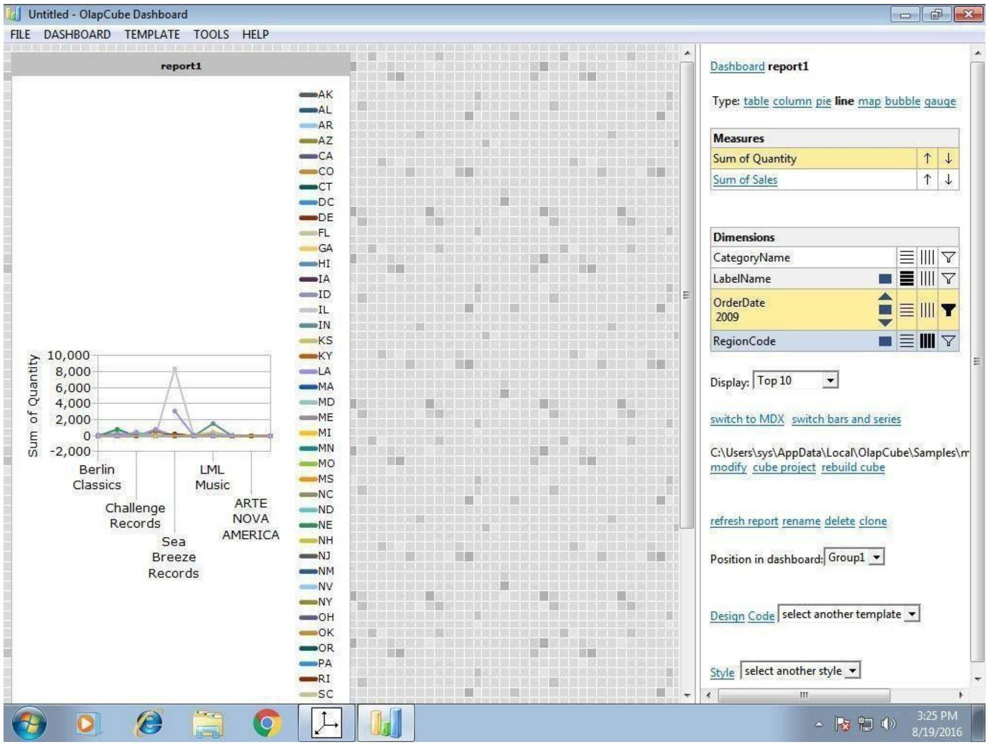
Display: Top 10

C:\Users\sys\AppData\Local\OlapCube\Samples\m

Position in dashboard: Group1

select another template

select another style



Display: Top 10

C:\Users\sys\AppData\Local\OlapCube\Samples\m

Position in dashboard: Group1

select another template

select another style

Exercise:

1. Apply the OLAP operations for the above banking application.

Record Notes

Experiment 5: Implementation of Apriori Algorithm

Description:

The Apriori algorithm is an influential algorithm for mining frequent item sets for Boolean association rules. It uses a "bottom-up" approach, where frequent subsets are extended one at a time (a step known as candidate generation, and groups of candidates are tested against the data).

- **Problem:**

TID	ITEMS
100	1,3,4
200	2,3,5
300	1,2,3,5
400	2,5

To find frequent item sets for above transaction with a minimum support of 2 having confidence measure of 70% (i.e, 0.7).

Procedure:

Step 1:

Count the number of transactions in which each item occurs

TID	ITEMS
1	2
2	3
3	3
4	1
5	3

Step 2:

Eliminate all those occurrences that have transaction numbers less than the minimum support (2 in this case).

ITEM	NO. OF TRANSACTIONS
1	2
2	3
3	3
5	3

This is the single items that are bought frequently. Now let's say we want to find a pair of items that are bought frequently. We continue from the above table (Table in step 2).

Step 3:

We start making pairs from the first item like 1,2;1,3;1,5 and then from second item like 2,3;2,5. We do not perform 2,1 because we already did 1,2 when we were making pairs with 1 and buying 1 and 2 together is same as buying 2 and 1 together. After making all the pairs we get,

ITEM PAIRS
1,2
1,3
1,5
2,3
2,5
3,5

Step 4:

Now, we count how many times each pair is bought together.

ITEM PAIRS	NO.OF TRANSACTIONS
1,2	1
1,3	2
1,5	1
2,3	2
2,5	3
3,5	2

Step 5:

Again remove all item pairs having number of transactions less than 2.

ITEM PAIRS	NO. OF TRANSACTIONS
1,3	2
2,3	2
2,5	3
3,5	2

These pair of items is bought frequently together. Now, let's say we want to find a set of three items that are bought together. We use above table (of step 5) and make a set of three items.

Step 6:

To make the set of three items we need one more rule (It is termed as self-join), it simply means, from item pairs in above table, we find two pairs with the same first numeric, so, we get (2,3) and (2,5), which gives (2,3,5). Then we find how many times (2, 3, 5) are bought together in the original table and we get the following

ITEM SET	NO. OF TRANSACTIONS
(2,3,5)	2

Thus, the set of three items that are bought together from this data are (2, 3, 5).

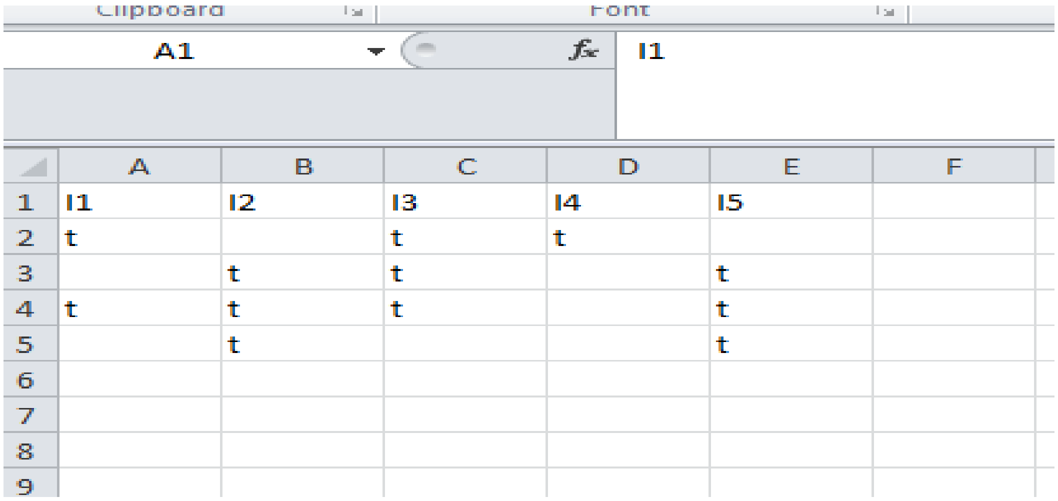
Confidence:

We can take our frequent item set knowledge even further, by finding association rules using the frequent item set. In simple words, we know (2, 3, 5) are bought together frequently, but what is the association between them. To do this, we create a list of all subsets of frequently bought items (2, 3, 5) in our case we get following subsets:

- {2}
- {3}
- {5}
- {2,3}
- {3,5}
- {2,5}

Now, we find association among all the subsets.
{2} => {3,5} : (If „2‟ is bought , what‟s the probability that „3‟ and „5‟ would be bought in same transaction)
Confidence = P (3□5□2)/ P(2) =2/3 =67%
{3}=>{2,5}= P (3□5□2)/P(3)=2/3=67%
{5}=>{2,3}= P (3□5□2)/P(5)=2/3=67%
{2,3}=>{5}= P (3□5□2)/P(2□3)=2/2=100%
{3,5}=>{2}= P (3□5□2)/P(3□5)=2/2=100%
{2,5}=>{3}= P (3□5□2)/ P(2□5)=2/3=67%
Also, considering the remaining 2-items sets, we would get the following associations-
{1}=>{3}=P(1□3)/P(1)=2/2=100%
{3}=>{1}=P(1□3)/P(3)=2/3=67%
{2}=>{3}=P(3□2)/P(2)=2/3=67%
{3}=>{2}=P(3□2)/P(3)=2/3=67%
{2}=>{5}=P(2□5)/P(2)=3/3=100%
{5}=>{2}=P(2□5)/P(5)=3/3=100%
{3}=>{5}=P(3□5)/P(3)=2/3=67%
{5}=>{3}=P(3□5)/P(5)=2?3=67%
Eliminate all those having confidence less than 70%. Hence, the rules would be –
{2,3}=>{5}, {3,5}=>{2}, {1}=>{3},{2}=>{5}, {5}=>{2}.

- Now these manual results should be checked with the rules generated in WEKA.

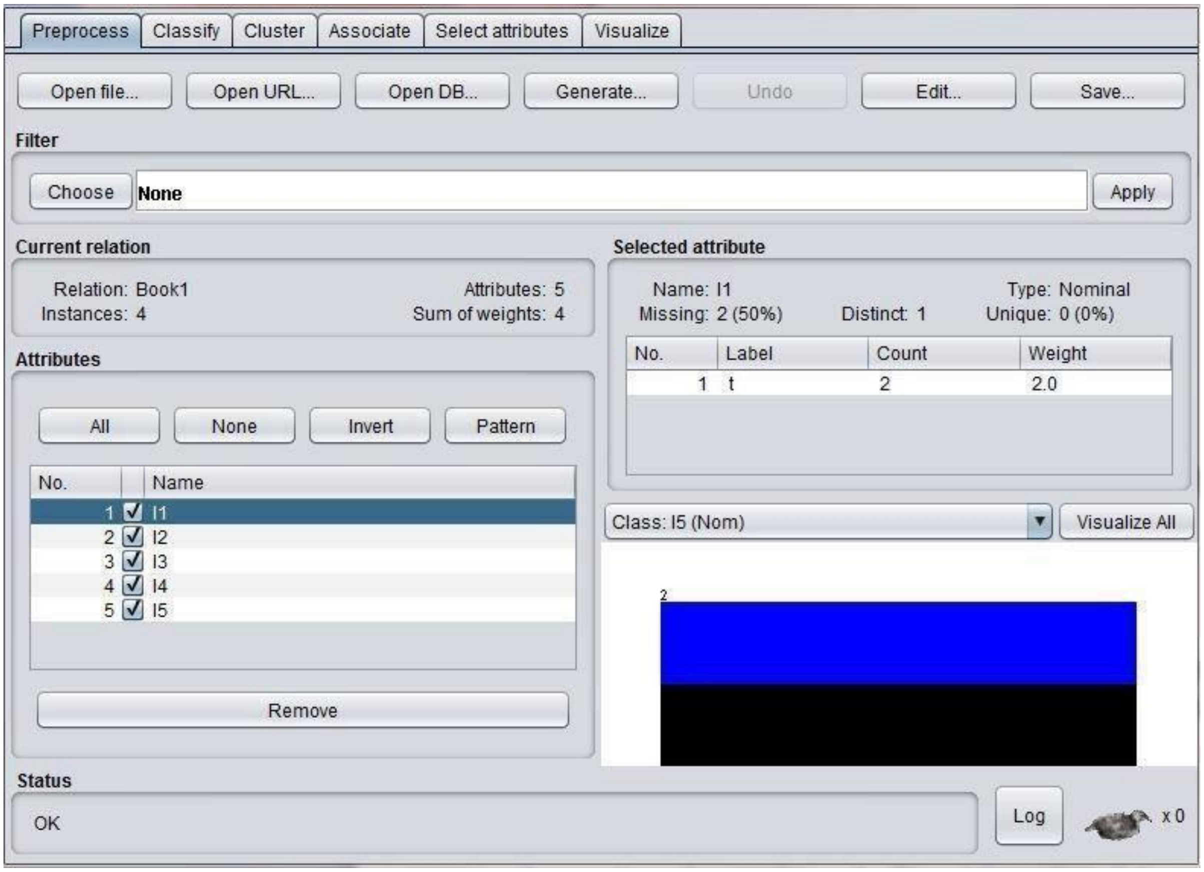


	A	B	C	D	E	F
1	I1	I2	I3	I4	I5	
2	t		t	t		
3		t	t		t	
4	t	t	t		t	
5		t			t	
6						
7						
8						
9						

So first create a csv file for the above problem, the csv file for the above problem will look like the rows and columns in the above figure. This file is written in excel sheet.

Procedure for running the rules in weka:
Step 1:

Open weka explorer and open the file and then select all the item sets. The figure gives a better understanding of how to do that.



Filter

None

Current relation

Relation: Book1 Attributes: 5
Instances: 4 Sum of weights: 4

Selected attribute

Name: I1 Type: Nominal
Missing: 2 (50%) Distinct: 1 Unique: 0 (0%)

No.	Label	Count	Weight
1	t	2	2.0

Attributes

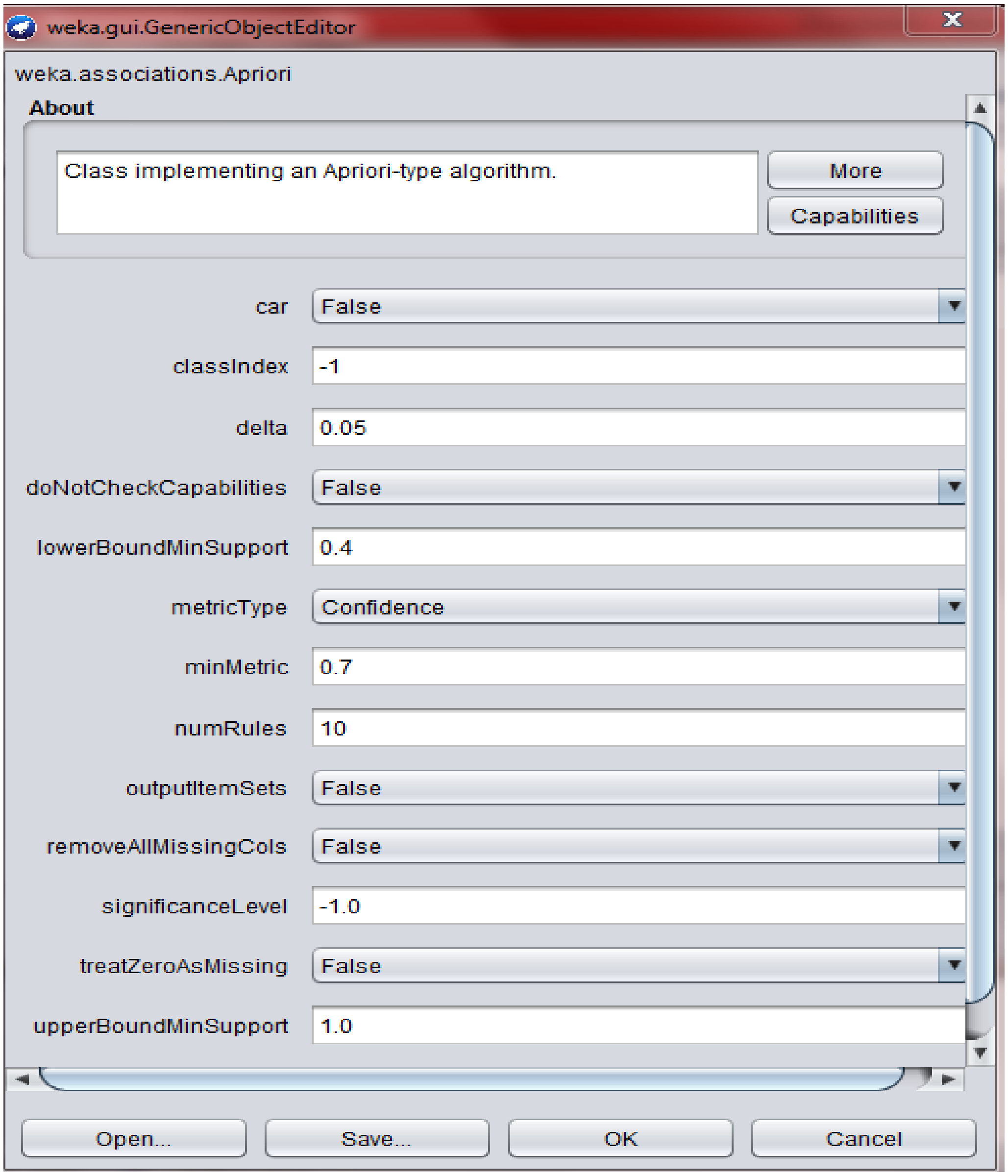
No.		Name
1	☑	I1
2	☑	I2
3	☑	I3
4	☑	I4
5	☑	I5

Class: I5 (Nom)

Status

OK

Step 2:
Now select the association tab and then choose apriori algorithm by setting the minimumsupport and confidence as shown in the figure



weka.gui.GenericObjectEditor

weka.associations.Apriori

About

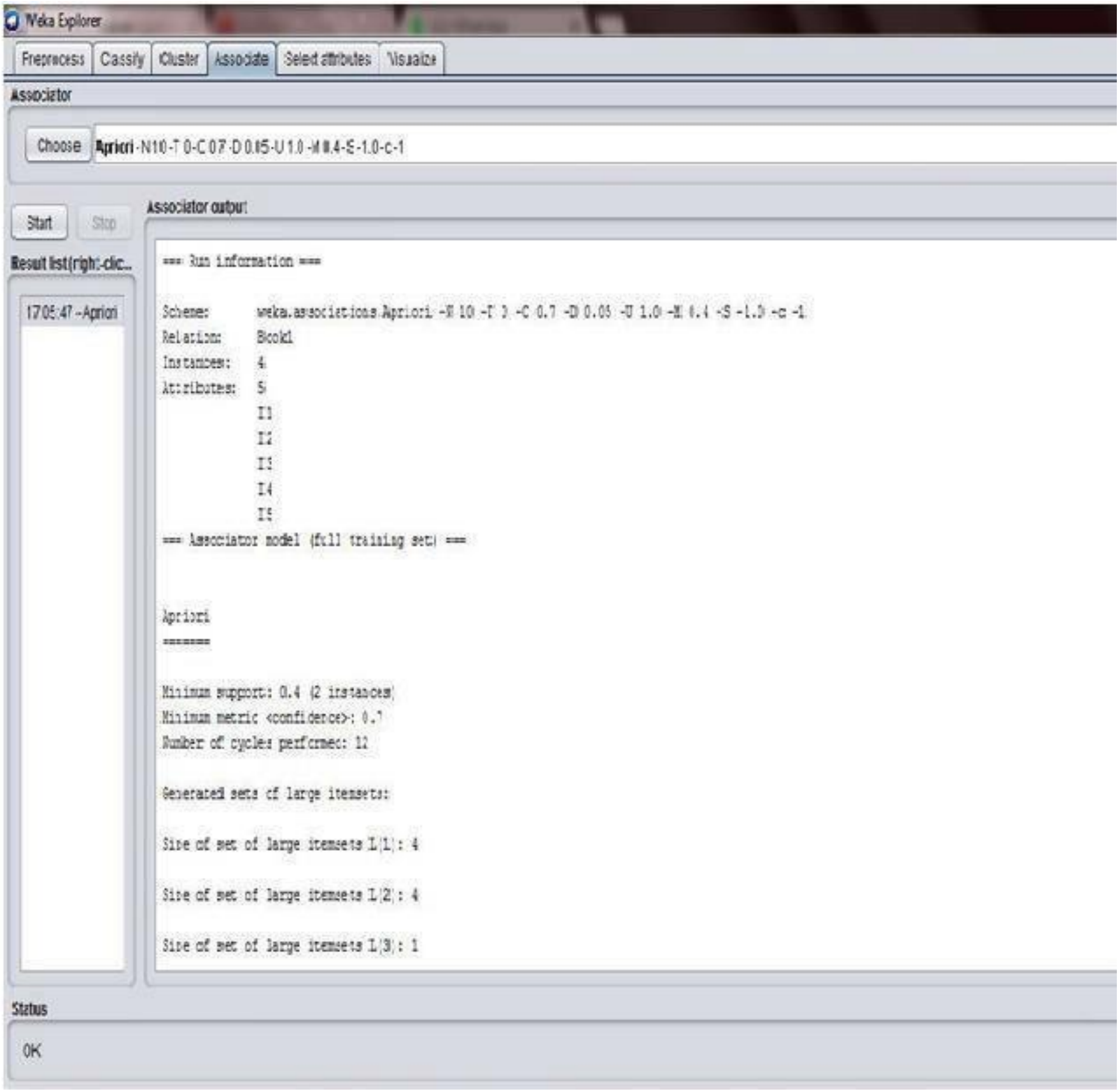
Class implementing an Apriori-type algorithm.

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.4
metricType	Confidence
minMetric	0.7
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0

Step 3:

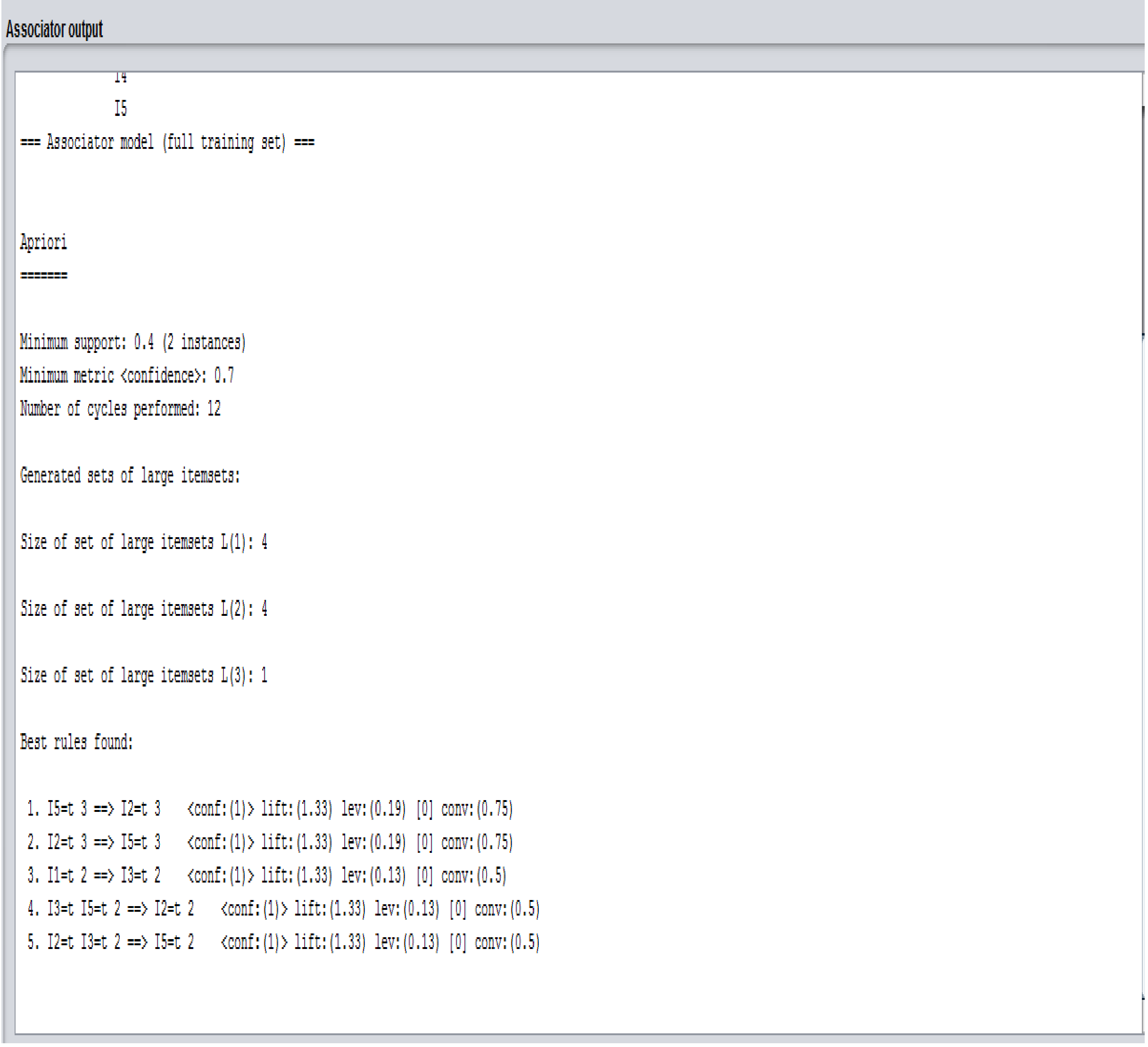
Now run the apriori algorithm with the set values of minimum support and the confidence. After running the weka generates the association rules and the respective confidence with minimum support as shown in the figure.

The above csv file has generated 5 rules as shown in the figure:



Apriori -N 10 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.4 -S -1.0 -c -1

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.4 -S -1.0 -c -1
Relation:     Book1
Instances:    4
Attributes:   5
              I1
              I2
              I3
              I4
              I5
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.4 (2 instances)
Minimum metric <confidence>: 0.7
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 4

Size of set of large itemsets L(3): 1
```



```
            I4
            I5
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.4 (2 instances)
Minimum metric <confidence>: 0.7
Number of cycles performed: 12

Generated sets of large itemsets:

Size of set of large itemsets L(1): 4

Size of set of large itemsets L(2): 4

Size of set of large itemsets L(3): 1

Best rules found:

 1. I5=t 3 ==> I2=t 3    <conf:(1)> lift:(1.33) lev:(0.19) [0] conv:(0.75)
 2. I2=t 3 ==> I5=t 3    <conf:(1)> lift:(1.33) lev:(0.19) [0] conv:(0.75)
 3. I1=t 2 ==> I3=t 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
 4. I3=t I5=t 2 ==> I2=t 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
 5. I2=t I3=t 2 ==> I5=t 2    <conf:(1)> lift:(1.33) lev:(0.13) [0] conv:(0.5)
```

Conclusion:
As we have seen the total rules generated by us manually and by the weka are matching, hence the rules generated are 5.

Exercise:

1.Apply the Apriori algorithm on Airport noise monitoring dataset discriminating between patients with parkin sons and neurological diseases using voice recording 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

(5a) Aim: To generate association rules using FP Growth Algorithm

PROBLEM:

To find all frequent item sets in following dataset using FP-growth algorithm. Minimum support=2 and confidence =70%

TID	ITEMS
100	1,3,4
200	2,3,5
300	1,2,3,5
400	2,5

Solution:

Similar to Apriori Algorithm, find the frequency of occurrences of all each item in dataset and then prioritize the items according to its descending order of its frequency of occurrence. Eliminating those occurrences with the value less than minimum support and assigning the priorities, we obtain the following table.

ITEM	NO. OF TRANSACTIONS	PRIORITY
1	2	4
2	3	1
3	3	2
5	3	3

Re-arranging the original table, we obtain

TID	ITEMS
100	1,3
200	2,3,5
300	2,3,5,1
400	2,5

Construction of tree:
Note that all FP trees have „null" node as the root node. So, draw the root node first and attach the items of the row 1 one by one respectively and write their occurrences in front of it. The tree is further expanded by adding nodes according to the prefixes (count) formed and by further incrementing the occurrences every time they occur and hence the tree is built.

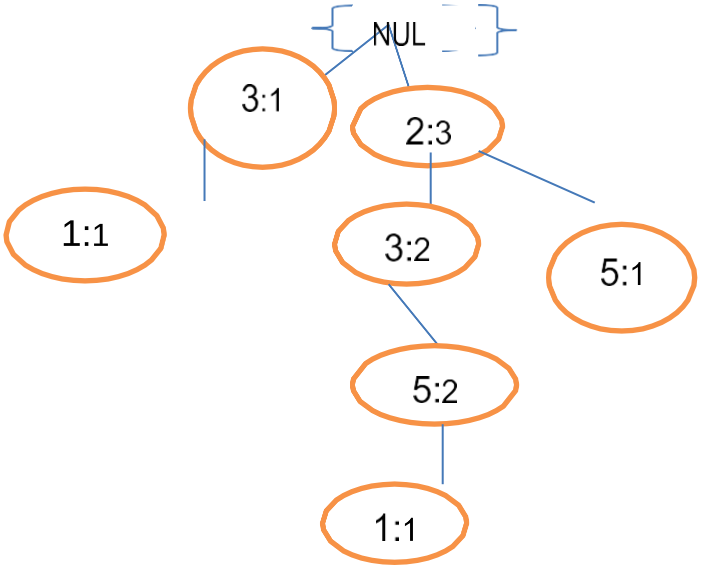
Prefixes:

- 1->3:1 2,3,5:1
- 5->2,3:2 2:1
- 3->2:2

Frequent item sets:

- 1-> 3:2 /*2 and 5 are eliminated because they"re less than minimum support, and the occurrence of 3 is obtained by adding the occurrences in both the instances*/
- Similarly, 5->2,3:2 ; 2:3;3:2
- 3->2 :2

Therefore, the frequent item sets are {3,1}, {2,3,5}, {2,5}, {2,3},{3,5}
The tree is constructed as below:



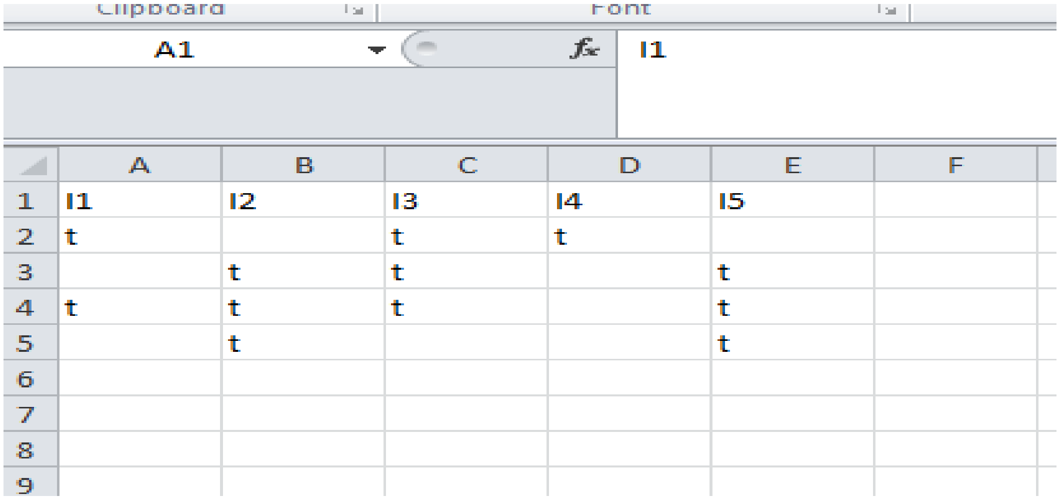
Generating the association rules for the following tree and calculating the confidence measures we get-

- {3}=>{1}=2/3=67%
- {1}=>{3}=2/2=100%
- {2}=>{3,5}=2/3=67%

- {2,5}=>{3}=2/3=67%
- {3,5}=>{2}=2/2=100%
- {2,3}=>{5}=2/2=100%
- {3}=>{2,5}=2/3=67%
- {5}=>{2,3}=2/3=67%
- {2}=>{5}=3/3=100%
- {5}=>{2}=3/3=100%
- {2}=>{3}=2/3=67%
- {3}=>{2}=2/3=67%

Thus eliminating all the sets having confidence less than 70%, we obtain the following conclusions:
{1}=>{3} , {3,5}=>{2} , {2,3}=>{5} , {2}=>{5}, {5}=>{2}.

As we see there are 5 rules that are being generated manually and these are to be checked against the results in WEKA. Inorder to check the results in the tool we need to follow the similar procedure like
Apriori.



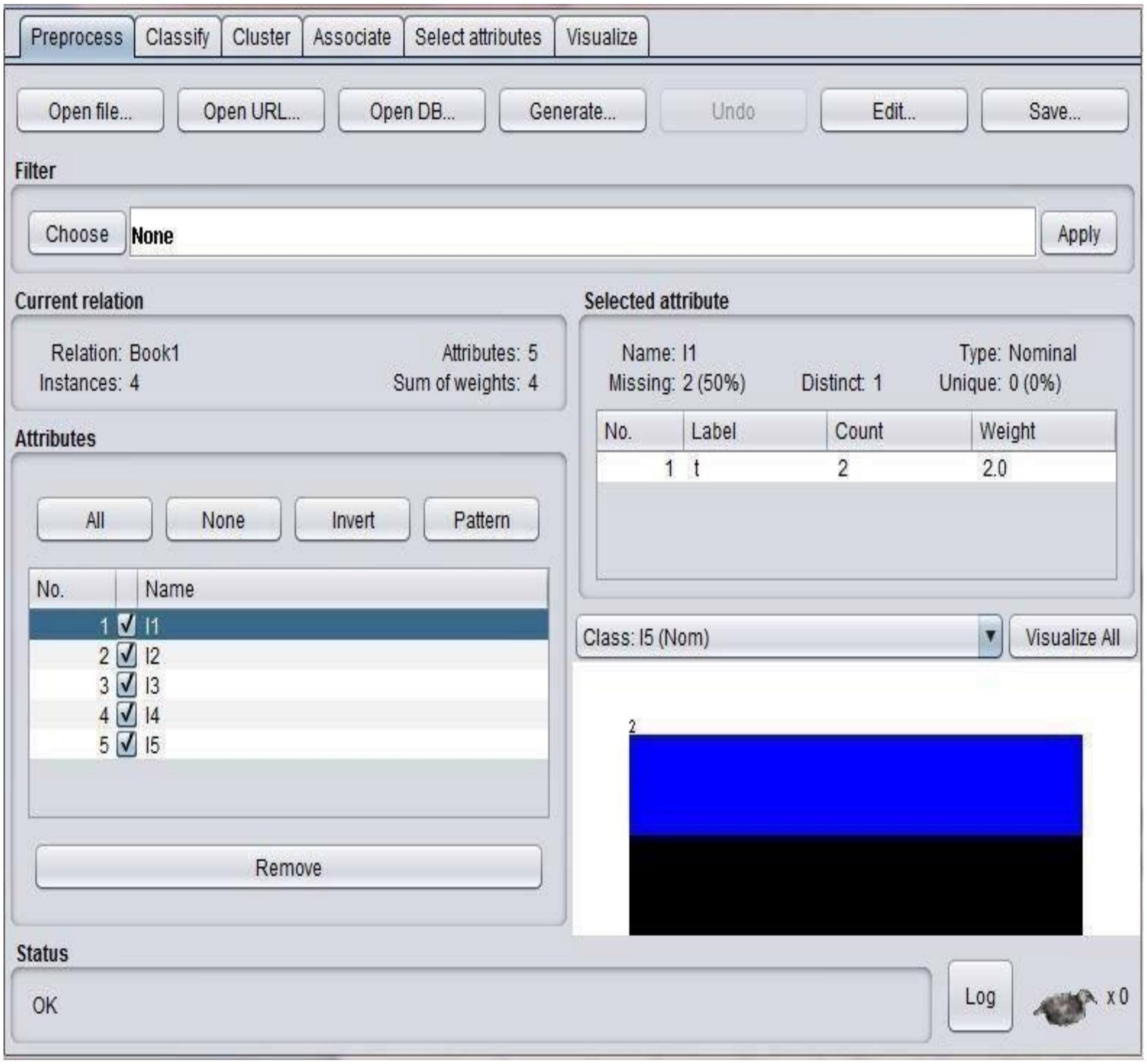
	A	B	C	D	E	F
1	I1	I2	I3	I4	I5	
2	t		t	t		
3		t	t		t	
4	t	t	t		t	
5		t			t	
6						
7						
8						
9						

So first create a csv file for the above problem, the csv file for the above problem will look like the rows and columns in the above figure. This file is written in excel sheet.

Procedure for running the rules in weka:

Step 1:

Open weka explorer and open the file and then select all the item sets. The figure gives a better understanding of how to do that.



Filter

None

Current relation

Relation: Book1 Attributes: 5

Instances: 4 Sum of weights: 4

Selected attribute

Name: I1 Type: Nominal

Missing: 2 (50%) Distinct: 1 Unique: 0 (0%)

No.	Label	Count	Weight
1	t	2	2.0

Attributes

No.		Name
1	☑	I1
2	☑	I2
3	☑	I3
4	☑	I4
5	☑	I5

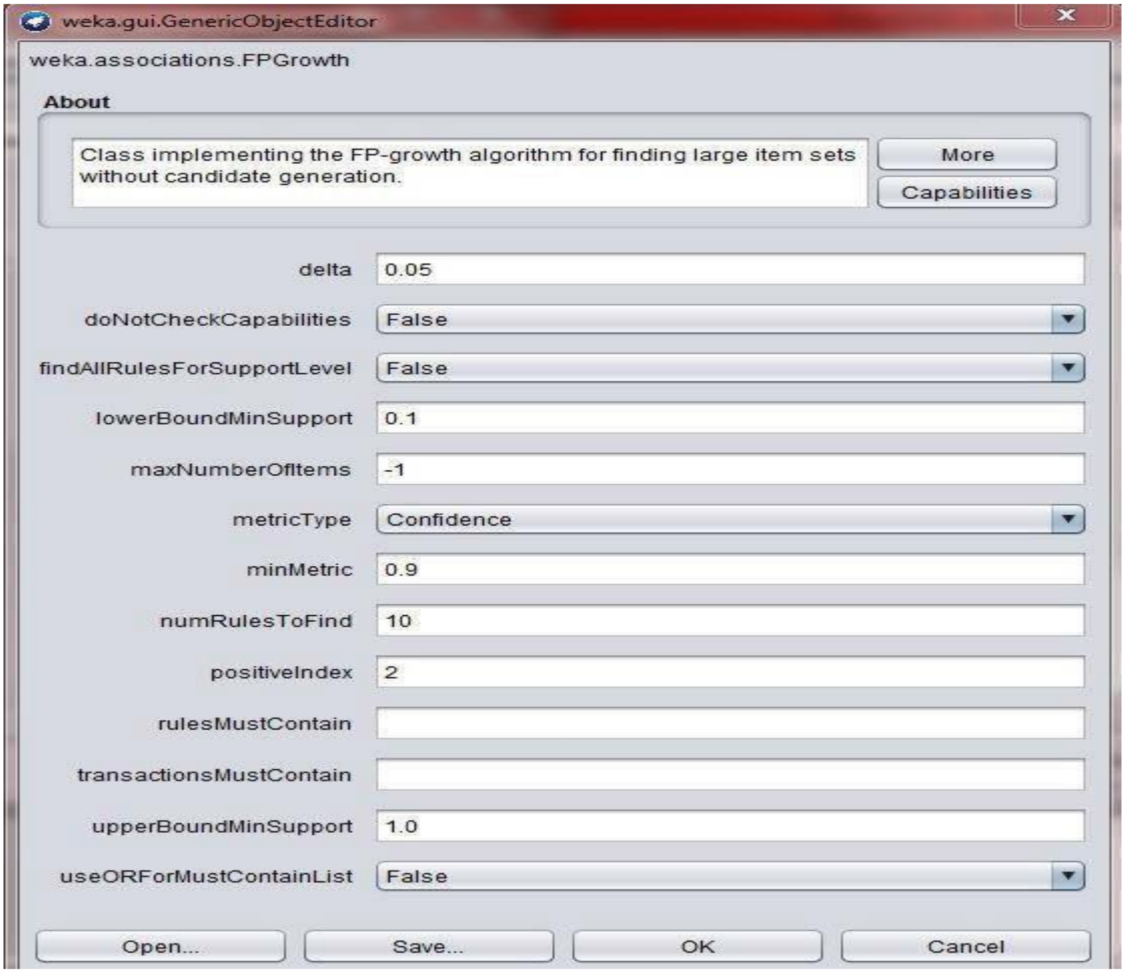
Class: I5 (Nom)

Status

OK

Step 2:

Now select the association tab and then choose FP growth algorithm by setting the minimum support and confidence as shown in the figure.



weka.gui.GenericObjectEditor

weka.associations.FPGrowth

About

Class implementing the FP-growth algorithm for finding large item sets without candidate generation.

delta 0.05

doNotCheckCapabilities False

findAllRulesForSupportLevel False

lowerBoundMinSupport 0.1

maxNumberOfItems -1

metricType Confidence

minMetric 0.9

numRulesToFind 10

positiveIndex 2

rulesMustContain

transactionsMustContain

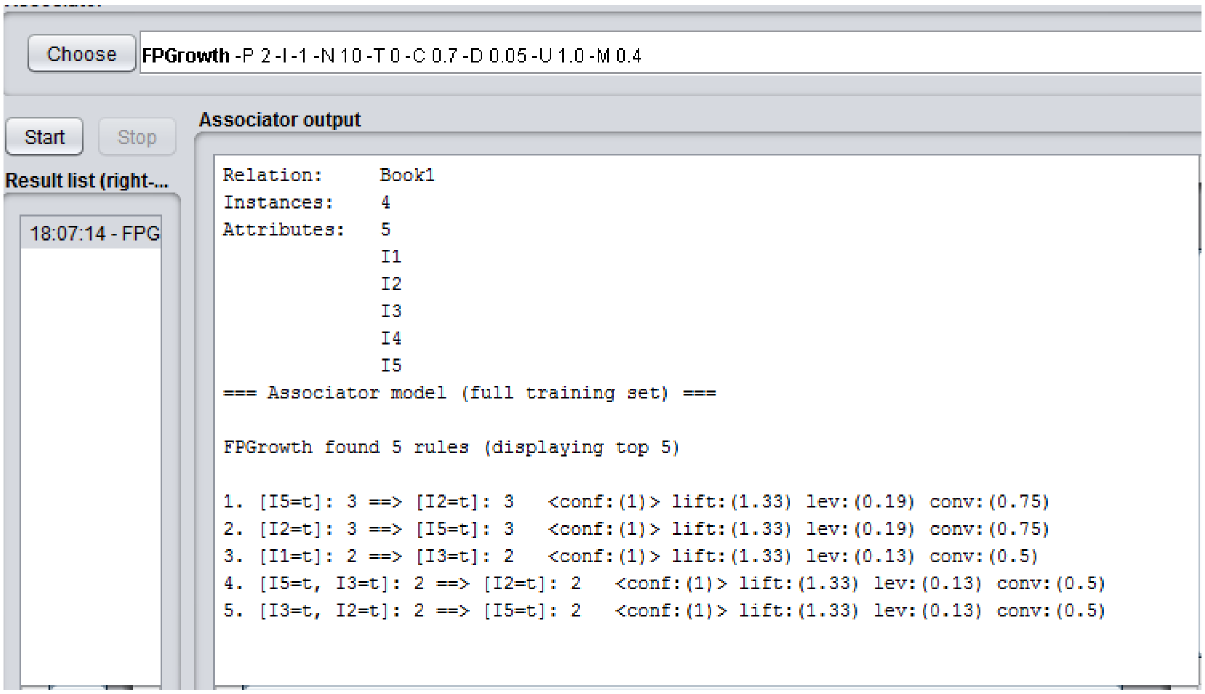
upperBoundMinSupport 1.0

useORForMustContainList False

Step 3:

Now run the FP Growth algorithm with the set values of minimum support and the confidence. After running the weka generates the association rules and the respective confidence with minimum support as shown in the figure.

The above csv file has generated 5 rules as shown in the figure:



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.4

```
Relation:     Book1
Instances:    4
Attributes:   5
              I1
              I2
              I3
              I4
              I5
=== Associator model (full training set) ===

FPGrowth found 5 rules (displaying top 5)

1. [I5=t]: 3 ==> [I2=t]: 3   <conf:(1)> lift:(1.33) lev:(0.19) conv:(0.75)
2. [I2=t]: 3 ==> [I5=t]: 3   <conf:(1)> lift:(1.33) lev:(0.19) conv:(0.75)
3. [I1=t]: 2 ==> [I3=t]: 2   <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
4. [I5=t, I3=t]: 2 ==> [I2=t]: 2   <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
5. [I3=t, I2=t]: 2 ==> [I5=t]: 2   <conf:(1)> lift:(1.33) lev:(0.13) conv:(0.5)
```

Conclusion:

As we have seen the total rules generated by us manually and by the weka are matching, hence the rules generated are 5.

Exercise

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

Record Notes

Experiment 7: Decision Tree Induction

Aim: Generate a Decision Tree by using J48 algorithm.

DESCRIPTION:

Decision tree learning is one of the most widely used and practical methods for inductive inference over supervised data. It represents a procedure for classifying categorical database on their attributes. This representation of acquired knowledge in tree form is intuitive and easy to assimilate by humans.

ILLUSTRATION:

Build a decision tree for the following data

AGE	INCOME	STUDENT	CREDIT_RATING	BUYS_COMPUTER
Youth	High	No	Fair	No
Youth	High	No	Excellent	No
Middle aged	High	No	Fair	Yes
Senior	Medium	No	Fair	Yes
Senior	Low	Yes	Fair	Yes
Senior	Low	Yes	Excellent	No
Middle aged	Medium	Yes	Excellent	Yes
Youth	Low	No	Fair	No
Youth	Medium	Yes	Fair	Yes
Senior	Medium	Yes	Fair	Yes
Youth	Medium	Yes	Excellent	Yes
Middle aged	Medium	No	Excellent	Yes
Middle aged	High	Yes	Fair	Yes
Senior	Medium	No	Excellent	No

The entropy is a measure of the uncertainty associated with a random variable. As uncertainty increases, so does entropy, values range from [0-1] to present the entropy of information

Entropy (D) = $\sum_{j=1}^{c} -p\log_2 p$
Information gain is used as an attribute selection measure; pick the attribute having the highest information gain, the gain is calculated by:
Gain (D, A) = Entropy (D) - $\sum_{j=1}^{c}(|\mathrm{Dj}|/|D|)Entropy(D)$
Where, D: A given data partition A: Attribute
V: Suppose we were partition the tuples in D on some attribute A having v distinct values D is split into v partition or subsets, (D1, D2….. Dj) , where Dj contains those tuples in D that have outcome Aj of A.

Class P: buys_computer=”yes”
Class N: buys_computer=”no”

Entropy (D) = -9/14log (9/14)-5/15log (5/14) =0.940
Compute the expected information requirement for each attribute start with the attribute age Gain (age, D)
= Entropy (D) - $\sum_{youth,middle-aged,senior}^{\dots}\left(\frac{Sv}{S}\right)\mathrm{Entropy}(Sv)$
= Entropy (D) – 5/14Entropy(Syouth)-4/14Entropy(Smiddle-aged)-5/14Entropy(Ssenior)
= 0.940-0.694
=0.246

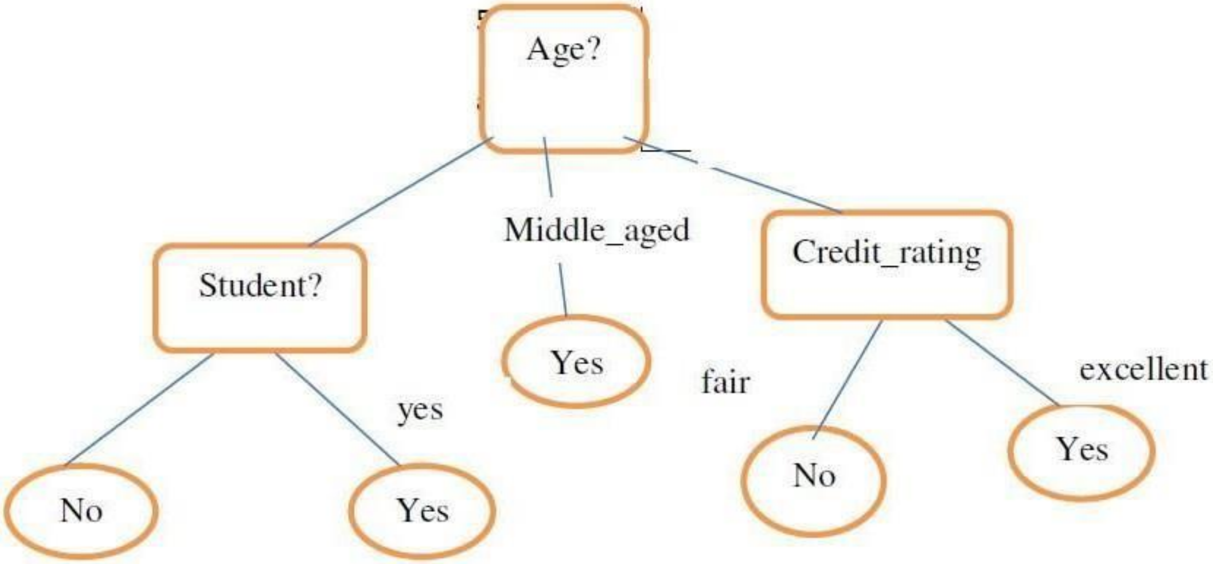
Similarly, for other attributes,
Gain (Income, D) =0.029
Gain (Student, D) = 0.151
Gain (credit_rating, D) = 0.048

Income	Student	Credit_rating	Class
High	No	Fair	No
High	No	Excellent	No
Medium	No	Fair	No
Low	Yes	Fair	Yes
medium	Yes	excellent	yes

Now, calculating information gain for subtable (age<=30)
I The attribute age has the highest information gain and therefore becomes the splitting
* attribute at the root node of the decision tree. Branches are grown for each outcome of age. These tuples are shown partitioned accordingly.

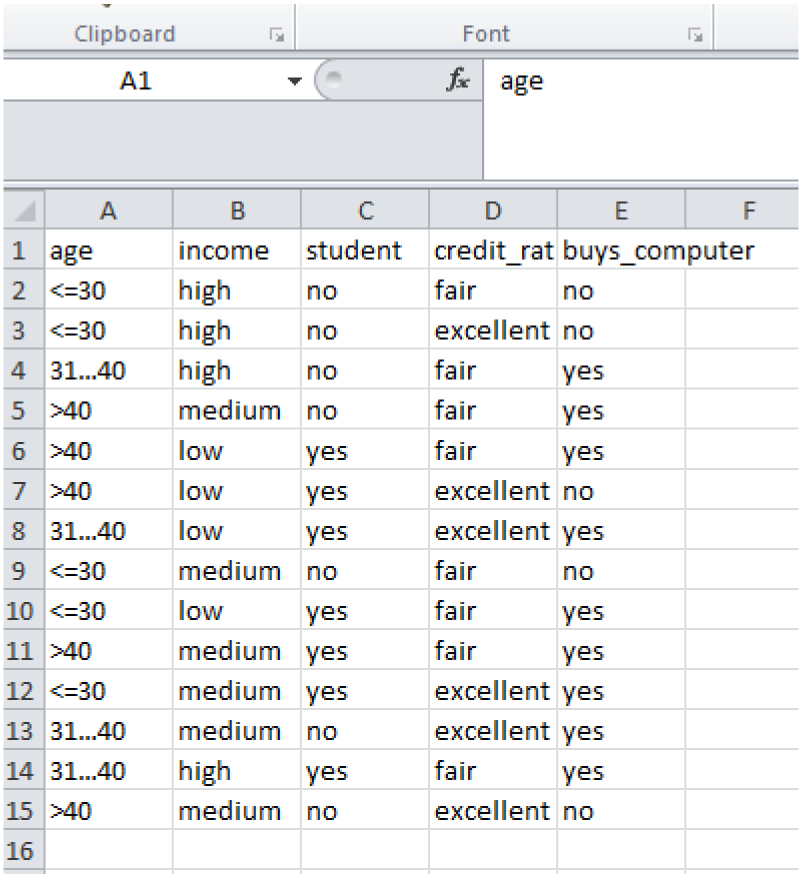
Income=”high” S11=0, S12=2
I=0
Income=”medium” S21=1 S22=1
I (S21, S23) = 1
Income=”low” S31=1 S32=0
I=0
Entropy for income
E(income) = (2/5)(0) + (2/5)(1) + (1/5)(0) = 0.4
Gain(income) = 0.971 - 0.4 = 0.571

Similarly, Gain(student)=0.971
Gain(credit)=0.0208
Gain(student) is highest ,



A decision tree for the concept buys_computer, indicating whether a customer at All Electronics is likely to purchase a computer. Each internal (non-leaf) node represents a test on an attribute. Each leaf node represents a class (either buys_computer=”yes” or buys_computer=”no”.

first create a csv file for the above problem,the csv file for the above problem will look like the rows and columns in the above figure. This file is written in excel sheet.



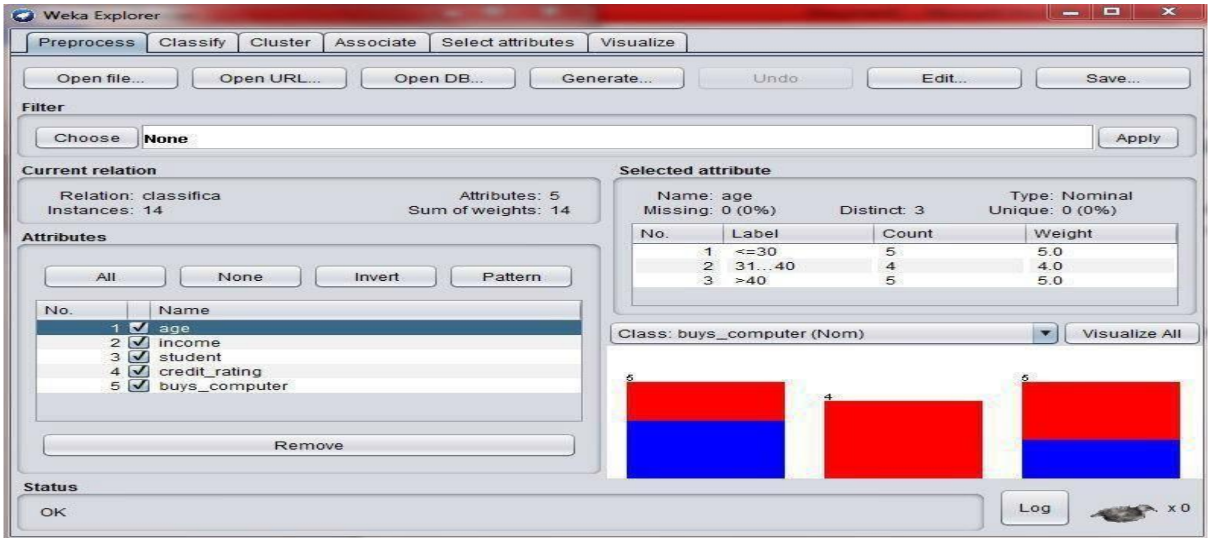
age

	A	B	C	D	E	F
1	age	income	student	credit_rat	buys_computer	
2	<=30	high	no	fair	no	
3	<=30	high	no	excellent	no	
4	31…40	high	no	fair	yes	
5	>40	medium	no	fair	yes	
6	>40	low	yes	fair	yes	
7	>40	low	yes	excellent	no	
8	31…40	low	yes	excellent	yes	
9	<=30	medium	no	fair	no	
10	<=30	low	yes	fair	yes	
11	>40	medium	yes	fair	yes	
12	<=30	medium	yes	excellent	yes	
13	31…40	medium	no	excellent	yes	
14	31…40	high	yes	fair	yes	
15	>40	medium	no	excellent	no	
16						

Procedure for running the rules in weka:

Step 1:

Open weka explorer and open the file and then select all the item sets. The figure gives a better understanding of how to do that.



Filter: None

Current relation
Relation: classifica
Instances: 14
Attributes: 5
Sum of weights: 14

No.	Name
1	☑ age
2	☑ income
3	☑ student
4	☑ credit_rating
5	☑ buys_computer

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

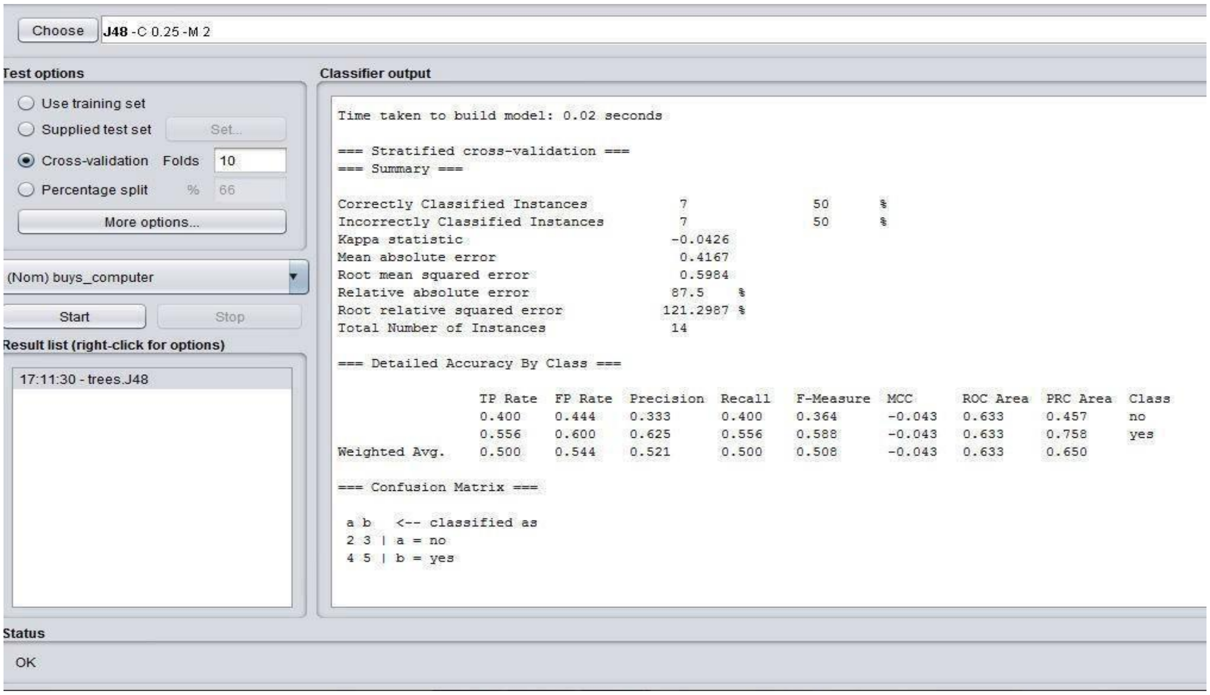
No.	Label	Count	Weight
1	<=30	5	5.0
2	31…40	4	4.0
3	>40	5	5.0

Class: buys_computer (Nom)

Status: OK

Step2:

Now select the classify tab in the tool and click on start button and then we can see the result of the problem as below



Choose J48 -C 0.25 -M 2

Test options

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

More options...

(Nom) buys_computer

Start   Stop

Result list (right-click for options)

17:11:30 - trees.J48

Classifier output

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.633     0.457     no
                 0.556    0.600    0.625      0.556    0.588      -0.043   0.633     0.758     yes
Weighted Avg.    0.500    0.544    0.521      0.500    0.508      -0.043   0.633     0.650

=== Confusion Matrix ===

 a b   <-- classified as
 2 3 | a = no
 4 5 | b = yes
```

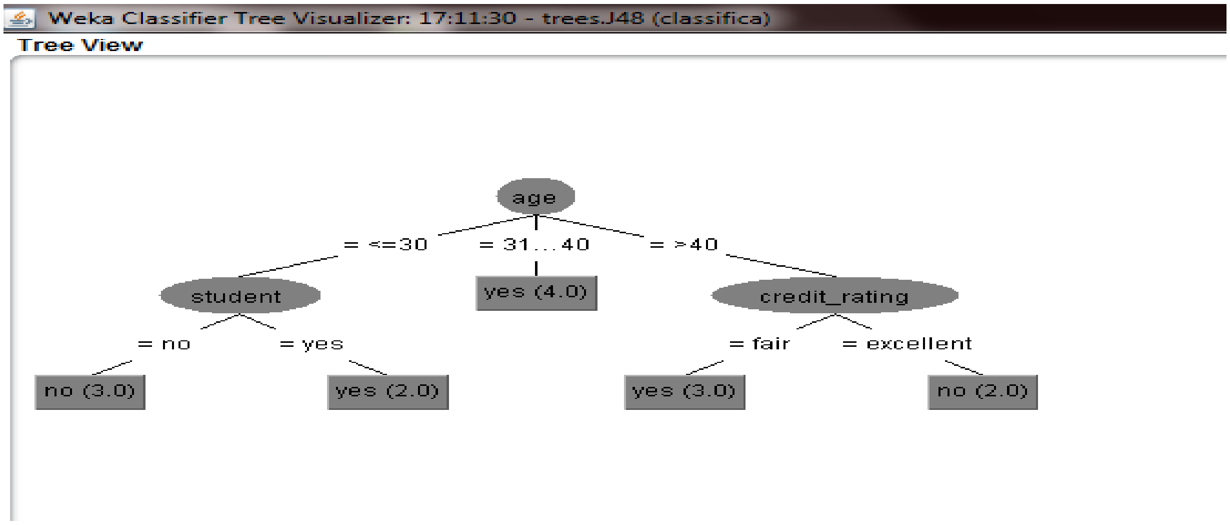
Status

OK

Step3:

Check the main result which we got manually and the result in weka by right clicking on the result and visualizing the tree.

The visualized tree in weka is as shown below:



Conclusion:
The solution what we got manually and the weka both are same.

Exercise:

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

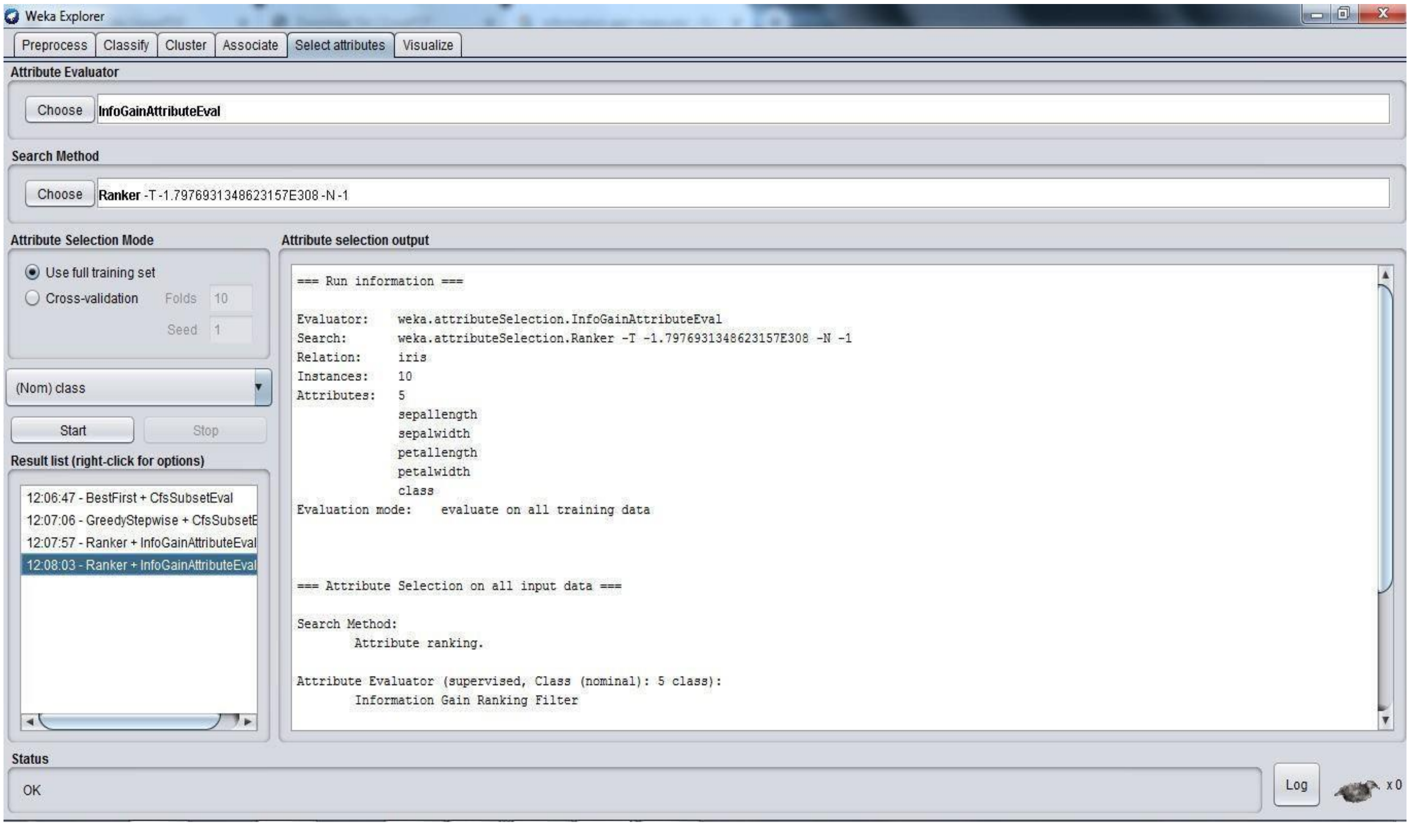
Record Notes

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



Attribute Evaluator

InfoGainAttributeEval

Search Method

Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10

Seed 1

(Nom) class

Result list (right-click for options)

12:06:47 - BestFirst + CfsSubsetEval
12:07:06 - GreedyStepwise + CfsSubsetE
12:07:57 - Ranker + InfoGainAttributeEval
12:08:03 - Ranker + InfoGainAttributeEval

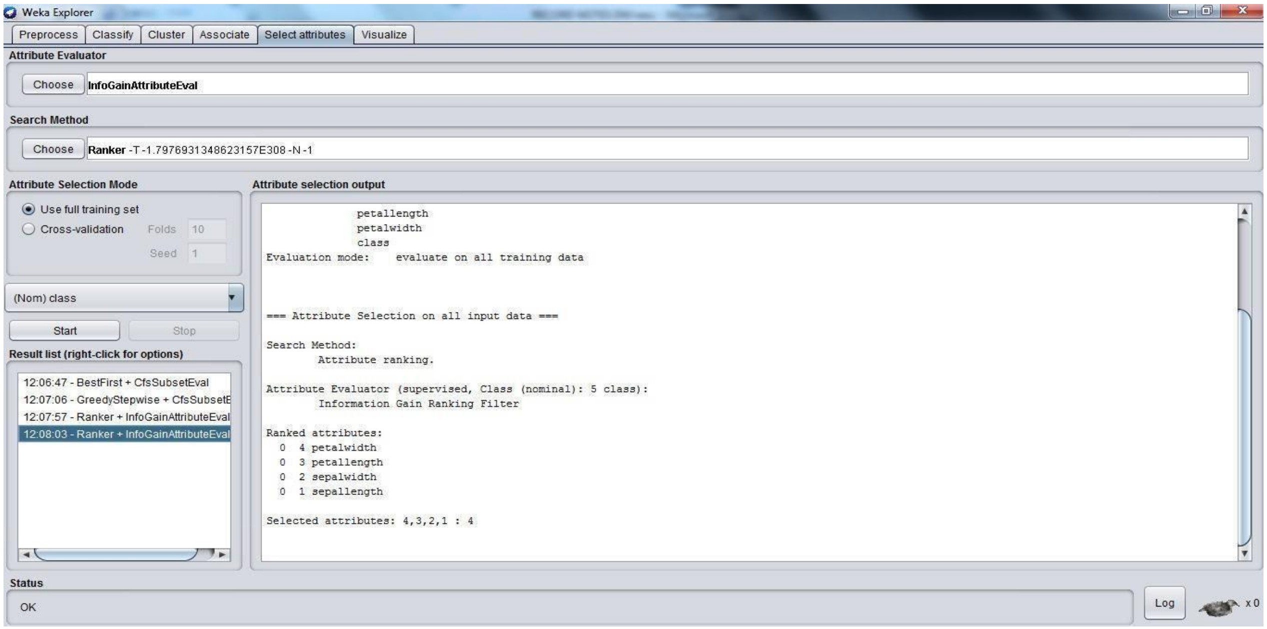
Attribute selection output

```
=== Run information ===

Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     iris
Instances:    10
Attributes:   5
              sepallength
              sepalwidth
              petallength
              petalwidth
              class
Evaluation mode:    evaluate on all training data



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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 class):
        Information Gain Ranking Filter
```

Status

OK

- Click on start button.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode

● Use full training set

○ Cross-validation Folds 10 Seed 1

(Nom) class

Result list (right-click for options)

- 12:06:47 - BestFirst + CfsSubsetEval
- 12:07:06 - GreedyStepwise + CfsSubsetE
- 12:07:57 - Ranker + InfoGainAttributeEval
- 12:08:03 - Ranker + InfoGainAttributeEval

Attribute selection output

```
              petallength
              petalwidth
              class
Evaluation mode:    evaluate on all training data



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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 class):
        Information Gain Ranking Filter

Ranked attributes:
  0  4 petalwidth
  0  3 petallength
  0  2 sepalwidth
  0  1 sepallength

Selected attributes: 4,3,2,1 : 4
```

Status: OK

Exercise

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

Record Notes

Experiment 9: Classification of data using Bayesian approach

AIM: To apply naïve bayes classifier on a given data set.

Description:

In machine learning, Naïve Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes‟ Theorem with strong (naïve) independence assumptions between the features

Example:

.

AGE	INCOME	STUDENT	CREDIT_RATING	BUYS_COMPUTER
<30	High	No	Fair	No
<30	High	No	Excellent	No
31-40	High	No	Fair	Yes
>40	Mediu m	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31-40	Mediu m	Yes	Excellent	Yes
<=30	Low	No	Fair	No
<=30	Mediu m	Yes	Fair	Yes
>40	Mediu m	Yes	Fair	Yes
<30	Mediu m	Yes	Excellent	Yes
31-40	Mediu m	No	Excellent	Yes
31-40	High	Yes	Fair	Yes
>40	Mediu m	No	Excellent	No

CLASS:

C1:buys_computer = 'yes'

C2:buys_computer='no'

DATA TO BECLASSIFIED :

X= (age<=30, income=Medium, Student=Yes, credit_rating=Fair)

- P(C1): P(buys_computer="yes")= 9/14 =0.643

  P (buys_computer="no") =5/14=0.357

- Compute P(X/C1) and p(x/c2) weget:

1. P(age="<=30" |buys_computer="yes")=2/9
2. P(age="<=30"|buys_computer="no")=3/5
3. P(income="medium"|buys_computer="yes")=4/9
4. P(income="medium"|buys_computer="no")=2/5
5. P(student="yes"|buys_computer="yes")=6/9
6. P(student="yes" |buys_computer="no")=1/5=0.2
7. P(credit_rating="fair "|buys_computer="yes")=6/9
8. P(credit_rating="fair" |buys_computer="no")=2/5

- X=(age<=30, income=medium, student=yes, credit_rating=fair) P(X/C1): P (X/buys_computer="yes")=2/9*4/9*6/9*6/9= 32/1134
  P(X/C2):P(X/buys_computer="no")=3/5*2/5*1 /5*2/5=12/125

  P(C1/X)=P(X/C1)*P(C1)

  P(X/buys_computer="yes")*P(buys_computer="yes")=(32/1134)*(9/14)=0.019

  P(C2/X)=p(x/c2)*p(c2)

  P (X/buys_computer="no")*P(buys_computer="no")=(12/125)*(5/14)=0.007

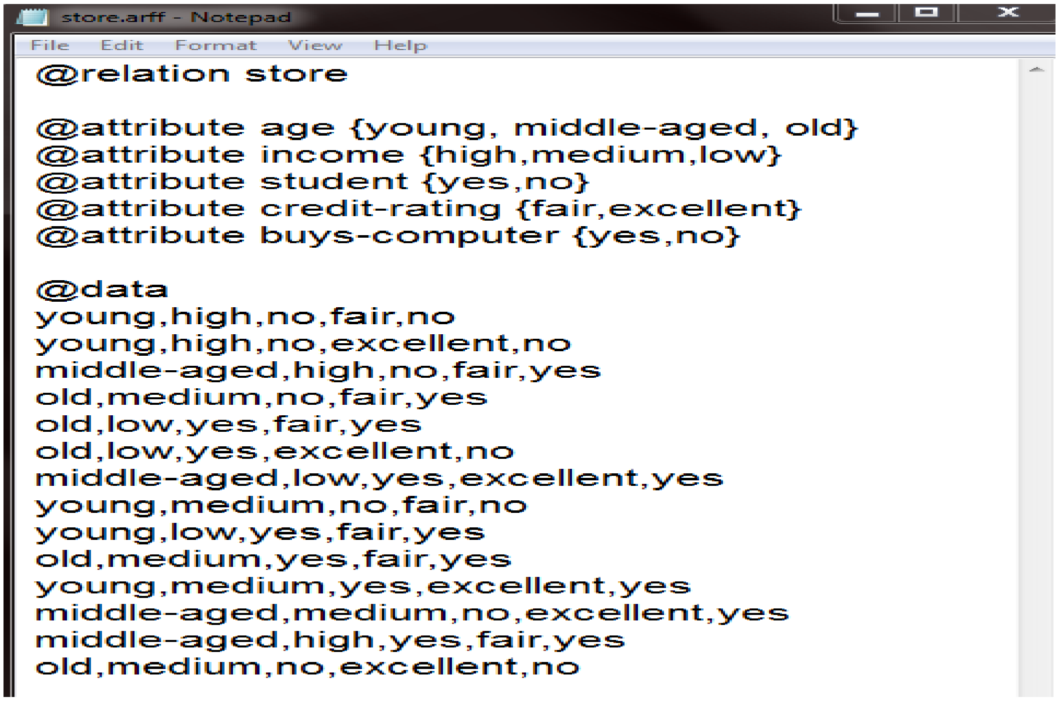
  Therefore, conclusion is that the given data belongs to C1 since P(C1/X)>P(C2/X)

Checking the result in the WEKA tool:

In order to check the result in the tool we need to follow a procedure.

Step 1:

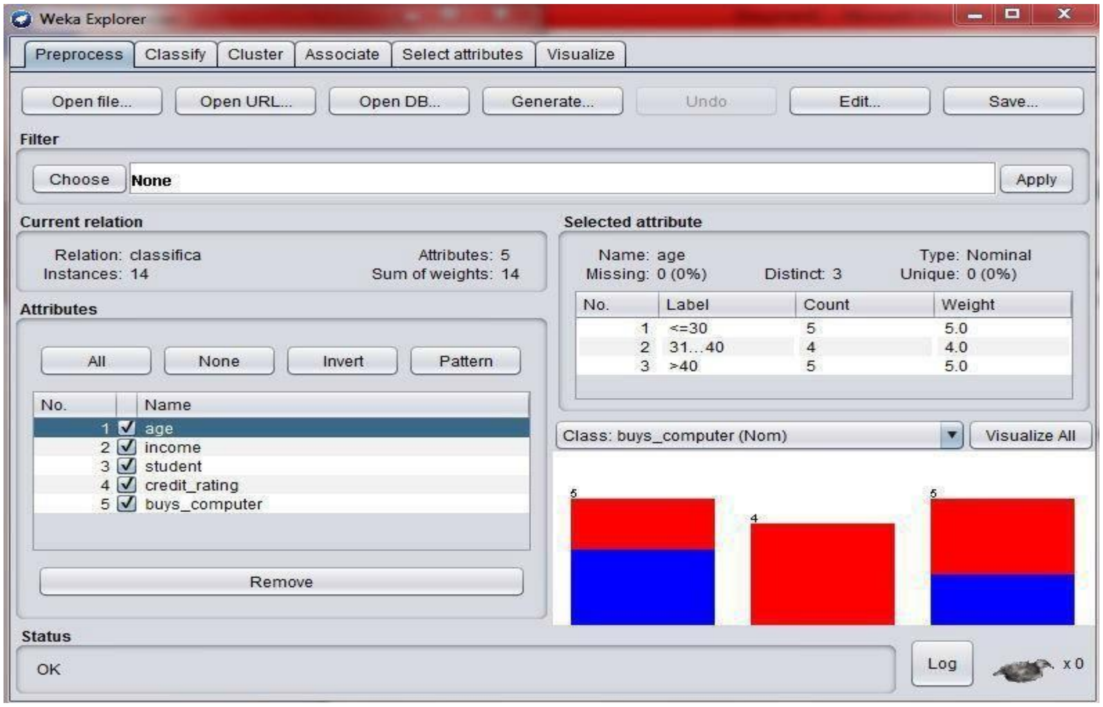
Create a csv file with the above table considered in the example. the arfffile will look as shown below:



```
@relation store

@attribute age {young, middle-aged, old}
@attribute income {high,medium,low}
@attribute student {yes,no}
@attribute credit-rating {fair,excellent}
@attribute buys-computer {yes,no}

@data
young,high,no,fair,no
young,high,no,excellent,no
middle-aged,high,no,fair,yes
old,medium,no,fair,yes
old,low,yes,fair,yes
old,low,yes,excellent,no
middle-aged,low,yes,excellent,yes
young,medium,no,fair,no
young,low,yes,fair,yes
old,medium,yes,fair,yes
young,medium,yes,excellent,yes
middle-aged,medium,no,excellent,yes
middle-aged,high,yes,fair,yes
old,medium,no,excellent,no
```

Step 2:

Now open weka explorer and then select all the attributes in the table.



Filter: None

Current relation

Relation: classifica   Attributes: 5
Instances: 14   Sum of weights: 14

Selected attribute

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

No.	Label	Count	Weight
1	<=30	5	5.0
2	31...40	4	4.0
3	>40	5	5.0

Attributes

No.		Name
1	☑	age
2	☑	income
3	☑	student
4	☑	credit_rating
5	☑	buys_computer

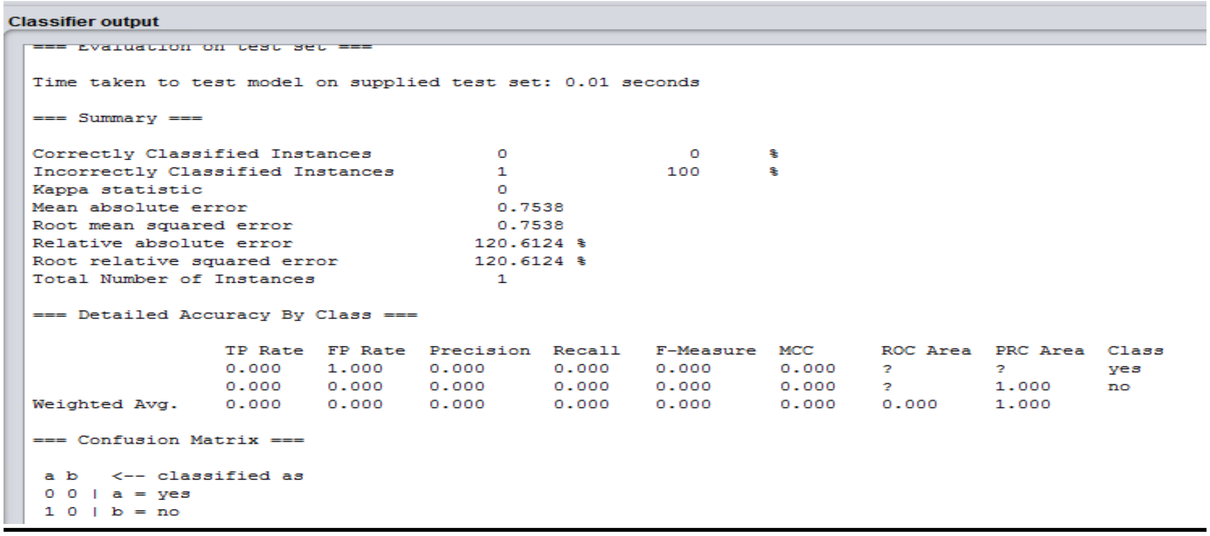
Class: buys_computer (Nom)

Status

OK

Step 3:

Select the classifier tab in the tool and choose baye‟s folder and then naïve baye‟s classifier to see the result as shown below.



```
=== Evaluation on test set ===

Time taken to test model on supplied test set: 0.01 seconds

=== Summary ===

Correctly Classified Instances           0                 0      %
Incorrectly Classified Instances         1               100      %
Kappa statistic                          0
Mean absolute error                      0.7538
Root mean squared error                  0.7538
Relative absolute error                120.6124 %
Root relative squared error            120.6124 %
Total Number of Instances                1

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.000    1.000    0.000      0.000    0.000      0.000    ?         ?         yes
                 0.000    0.000    0.000      0.000    0.000      0.000    ?         1.000     no
Weighted Avg.    0.000    0.000    0.000      0.000    0.000      0.000    0.000     1.000

=== Confusion Matrix ===

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

Exercise

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

Record Notes

Experiment 10: Implementation of K-means algorithm

DESCRIPTION:

K-means algorithm aims to partition n observations into “k clusters” in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in partitioning of the data into Voronoi cells.

ILLUSTRATION:

As a simple illustration of a k-means algorithm, consider the following data set consisting of the scores of two variables on each of the five variables.

I	X1	X2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5

This data set is to be grouped into two clusters: As a first step in finding a sensible partition, let the A & C values of the two individuals furthest apart (using the Euclidean distance measure), define the initial cluster means, giving:

Cluster	Individual	Mean Vector(Centroid)
Cluster1	A	(1,1)
Cluster2	C	(0,2)

The remaining individuals are now examined in sequence and allocated to the cluster to which they are closest, in terms of Euclidean distance to the cluster mean. The mean vector is recalculated each time a new member is added. This leads to the following series of steps:

	A	C
A	0	1.4
B	1	2.5
C	1.4	0
D	3.2	2.82
E	4.5	4.2

Initial partitions have changed, and the two clusters at this stage having the following characteristics.

	Individual	Mean vector(Centroid)
Cluster 1	A,B	(1,0.5)
Cluster 2	C,D,E	(1.7,3.7)

But we cannot yet be sure that each individual has been assigned to the right cluster. So, we compare each individual‟s distance to its own cluster mean and to that of the opposite cluster. And, we find:

I	A	C
A	0.5	2.7
B	0.5	3.7
C	1.8	2.4
D	3.6	0.5
E	4.9	1.9

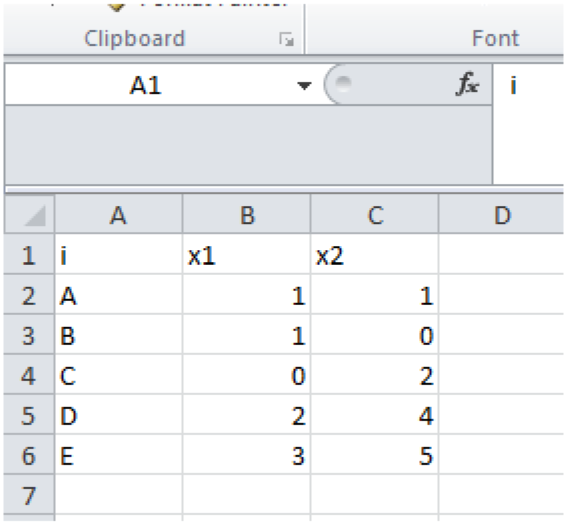
The individuals C is now relocated to Cluster 1 due to its less mean distance with the centroid points. Thus, its relocated to cluster 1 resulting in the new partition

	Individual	Mean vector(Centroid)
Cluster 1	A,B,C	(0.7,1)
Cluster 2	D,E	(2.5,4.5)

The iterative relocation would now continue from this new partition until no more relocation occurs. However, in this example each individual is now nearer its own cluster mean than that of the other cluster and the iteration stops, choosing the latest partitioning as the final cluster solution. Also, it is possible that the k-means algorithm won‟t find a final solution. In this case, it would be a better idea to consider stopping the algorithm after a pre-chosen maximum number of iterations.

Checking the solution in weka:
In order to check the result in the tool we need to follow a procedure.

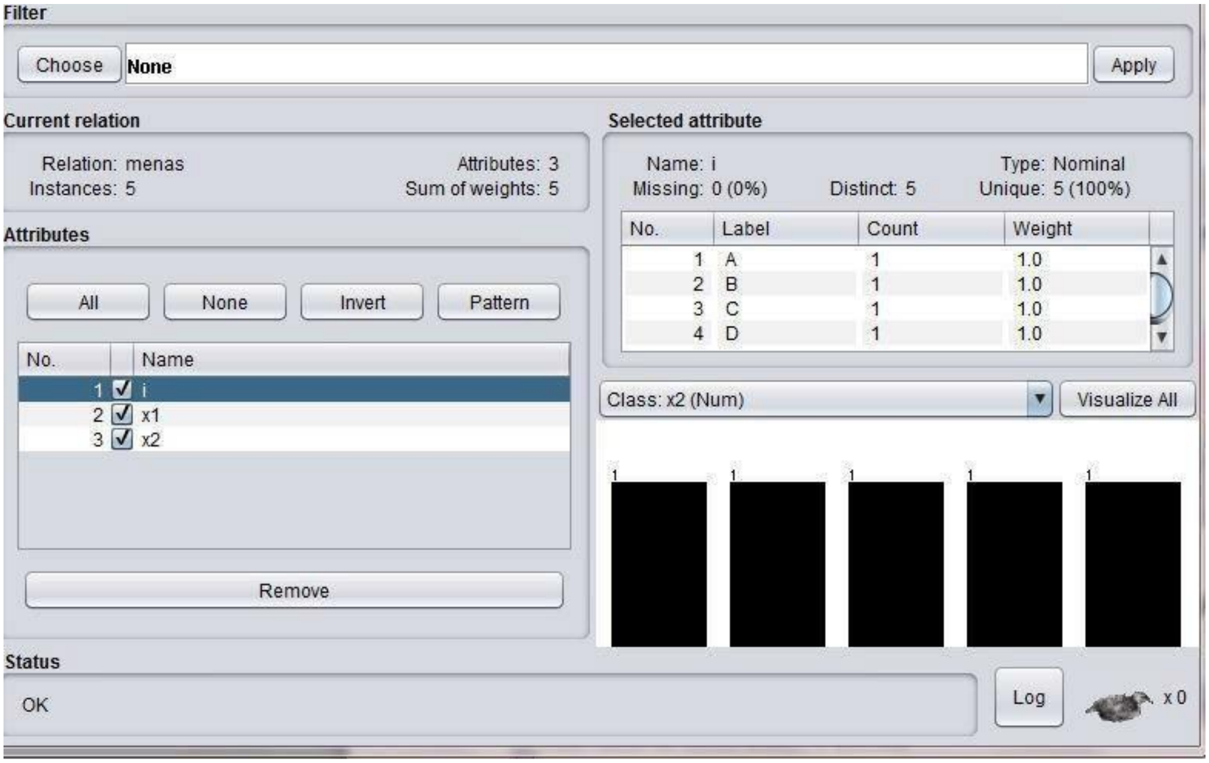
Step 1:
Create a csv file with the above table considered in the example. the csv file will look as shown below:



	A	B	C	D
1	i	x1	x2	
2	A	1	1	
3	B	1	0	
4	C	0	2	
5	D	2	4	
6	E	3	5	
7				

Step 2:

Now open weka explorer and then select all the attributes in the table.



Filter: None

Relation: menas Attributes: 3
Instances: 5 Sum of weights: 5

Name: i Type: Nominal
Missing: 0 (0%) Distinct: 5 Unique: 5 (100%)

No.	Label	Count	Weight
1	A	1	1.0
2	B	1	1.0
3	C	1	1.0
4	D	1	1.0

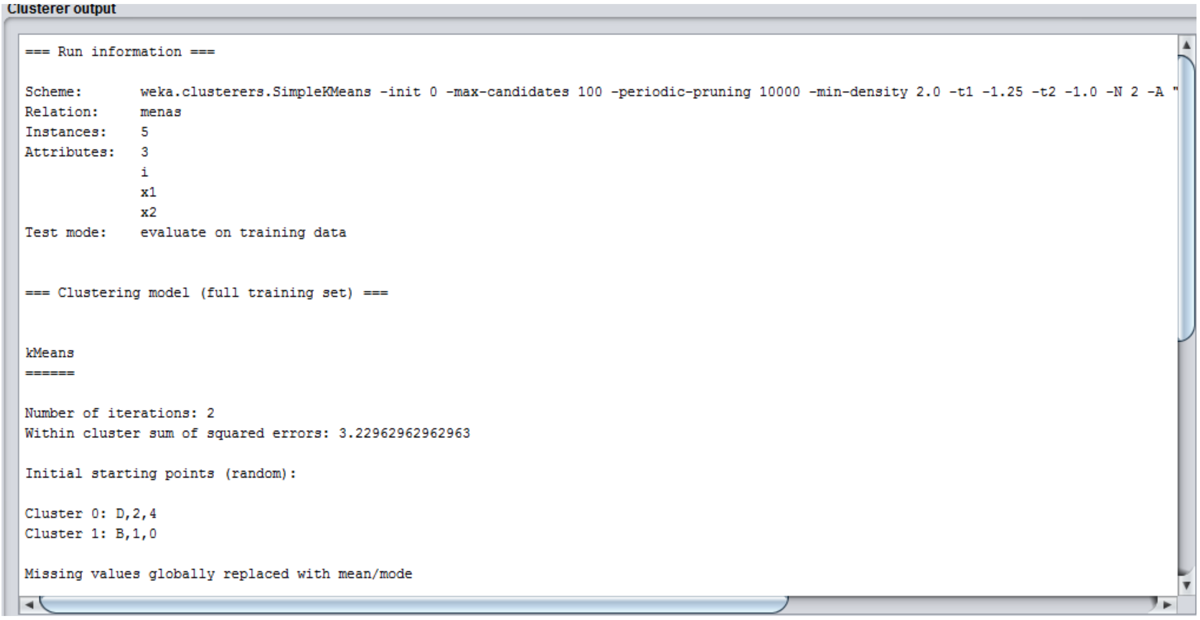
No.		Name
1	☑	i
2	☑	x1
3	☑	x2

Class: x2 (Num)

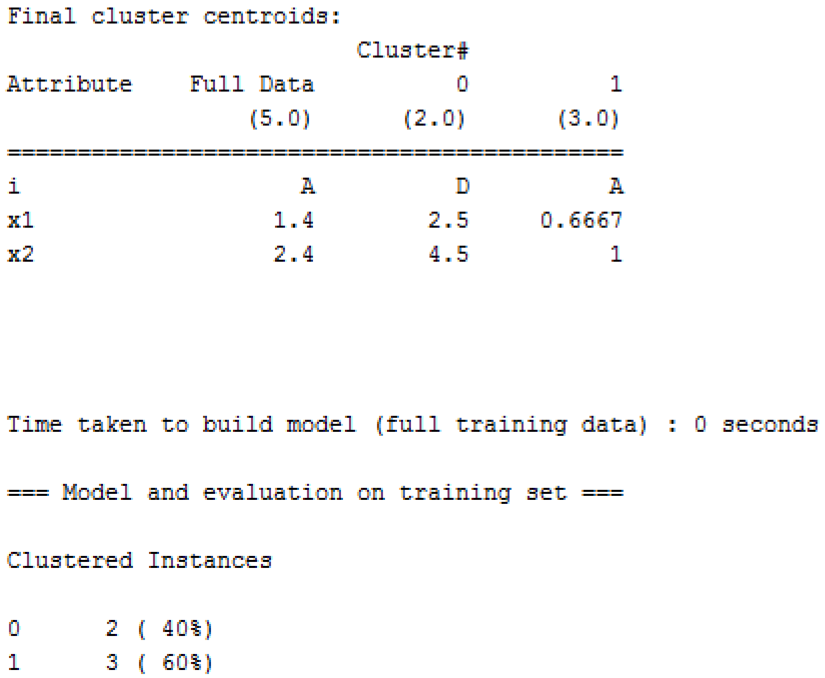
OK

Step 3:

Select the cluster tab in the tool and choose normal k-means technique to see the result as shown below.



```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "
Relation:     menas
Instances:    5
Attributes:   3
              i
              x1
              x2
Test mode:    evaluate on training data


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


kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 3.22962962962963

Initial starting points (random):

Cluster 0: D,2,4
Cluster 1: B,1,0

Missing values globally replaced with mean/mode
```



```
Final cluster centroids:
                         Cluster#
Attribute    Full Data          0          1
                 (5.0)      (2.0)      (3.0)
============================================
i                    A          D          A
x1                 1.4        2.5     0.6667
x2                 2.4        4.5          1




Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0      2 ( 40%)
1      3 ( 60%)
```

Exercise

1. Implement of K-means clustering using crime dataset.

Record Notes

Case Study

1. Create Placement.arff file to identify the students who are eligible for placements using KNN