

SVERI's College of Engineering, Pandharpur
Department of Computer Science and Engineering



Lab Manual of
DATA MINING (DM)
LY B. Tech (CSE) Sem-I

Prepared by:

Prof. S. M. SHINDE

VISION	
Institute Vision	Department Vision
To be recognized among the best institute in India for excellence in technical education.	To be nationally recognized for excellence in education augmented by research in the field of Computer Science and Engineering.
MISSION	
Institute Mission	Department Mission
To impart value added technical education through ambience of academic excellence, research and life skills by inculcating personal touch and respect in relationship amongst the stakeholders.	<ol style="list-style-type: none">1) To impart value-based education in Computer Science & Engineering, through effective teaching and learning approaches.2) To create ambience for academic excellence through fruitful interaction among various stakeholders.3) To inculcate best practices for innovative research, competitive employability and sustainable entrepreneurship development.
PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)	
The Department of Computer Science and Engineering has as its PEOs to produce graduate who: <ol style="list-style-type: none">1. Apply the Computer Science domain specific knowledge and skills in the growing software and related industries.2. Demonstrate leadership, professional ethics, project management and finance related attributes as employees or employers.3. Engage in life-long learning for professional advancement to develop innovative solutions for individual or societal problems.4. Demonstrate strong communication skills and ability to function effectively as an individual and part of a team.	
PROGRAMME SPECIFIC OUTCOMES (PSOs)	
Engineering Graduates will be able to: <ol style="list-style-type: none">1. Understand & design computer system using knowledge of Digital Techniques, Micro-Processor, Computer Organization, Advanced Computer Architecture, Operating System, System Programming, Compiler Construction, Application Softwares, etc.2. Interpret, analyze and design software system programming knowledge using Algorithmic Skills, Web Technology, Big Data Analytics, Networking Fundamentals, Machine Learning and Internet of Things.3. Adopt applications in emerging fields of Computer Science & Engineering.	

--

PROGRAMME OUTCOMES (POs)

Students graduating from Computer Science and Engineering will demonstrate:

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	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.
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 multidisciplinary 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.

Course Outcome (CO):

SR. NO	COURSE OBJECTIVES	BLOOMS LEVEL
CS415B.1	Identify the types of the data to be mined for a particular application.	BL-1
CS415B.2	Explain preprocessing statistical methods for any given raw data.	BL-2
CS415B.3	Apply proper data mining algorithms to build analytical applications.	BL-3
CS415B.4	Explain the roles that data mining plays in various fields and manipulate different data mining techniques.	BL-2
CS415B.5	Apply a wide range of Clustering, Classification and association rule mining algorithms.	BL-3
CS415B.6	Analysis of working of web crawlers, usage of web and outlier detection.	BL-4

EXPERIMENT LIST

Expt. No.	Experiment Title	CO	BL	PI
1)	Explore the WEKA Tool for Data Mining Techniques	CS415B.1	BL3	5.4.1
2)	Creating and the demonstration of the Arff File.	CS415B.1	BL3	4.4.2
3)	Demonstration of Pre-Processing Techniques on a given Data Sets - Pre-process a given dataset based on Handling Missing Values	CS415B.2	BL3	4.4.2
4)	Demonstration of classification rule process on a given dataset.	CS415B.3	BL3	4.4.2
5)	Demonstration of Association Rules using the Apriori Algorithm	CS415B.3	BL3	4.4.2
6)	Demonstration of association rules using fp growth algorithm.	CS415B.4	BL3	4.6.1
7)	Demonstration of Naïve bayes classification on a given data set	CS415B.5	BL3	4.6.1
8)	Demonstration of k-means clustering on a given data set	CS415B.5	BL3	4.6.1

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 user's 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:



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.

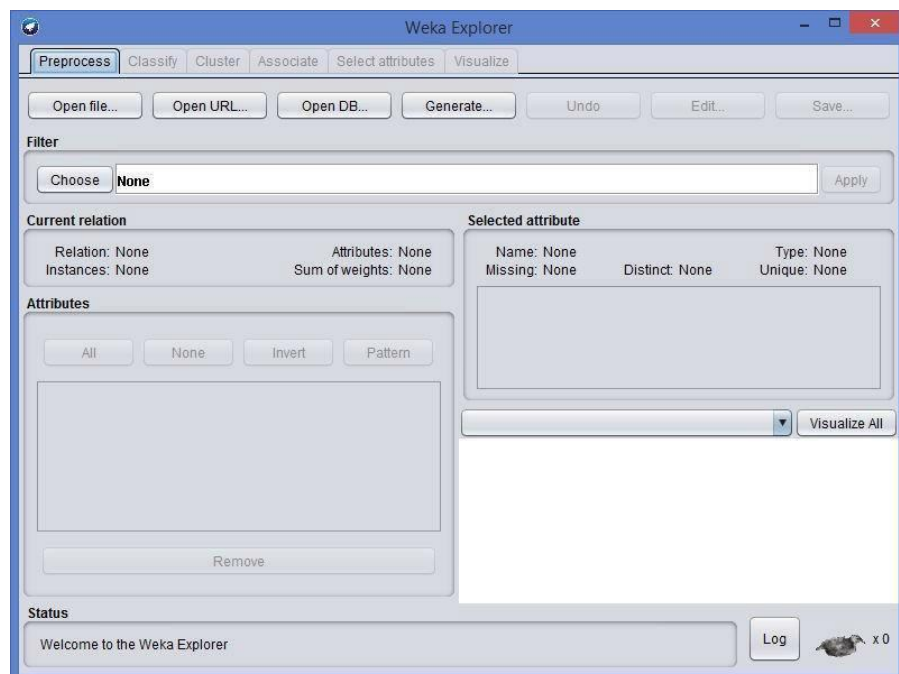
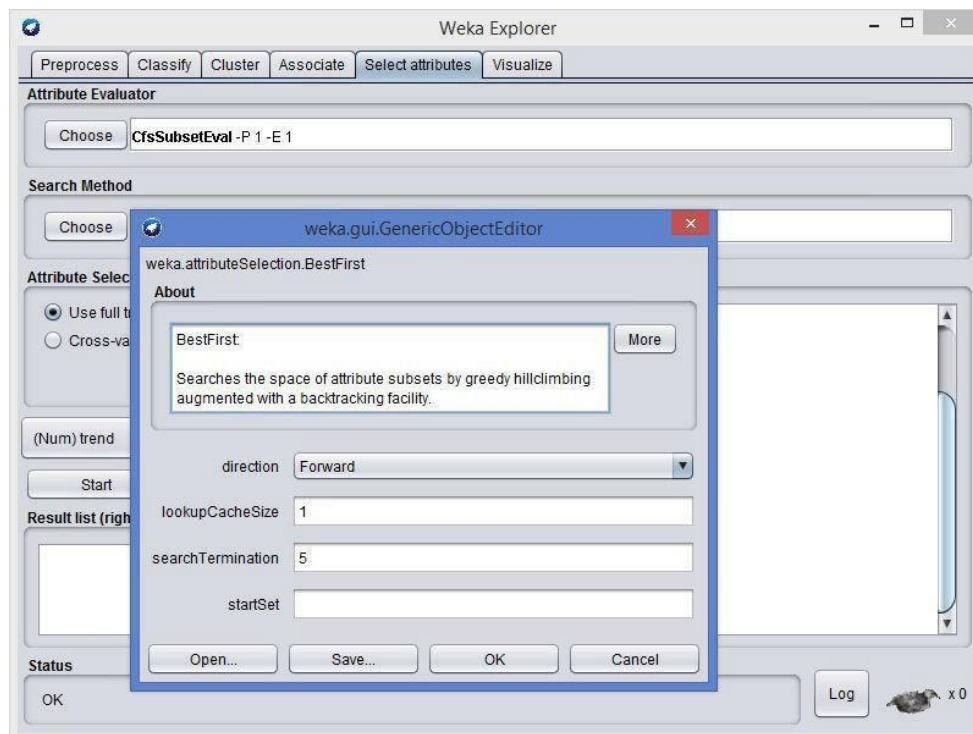
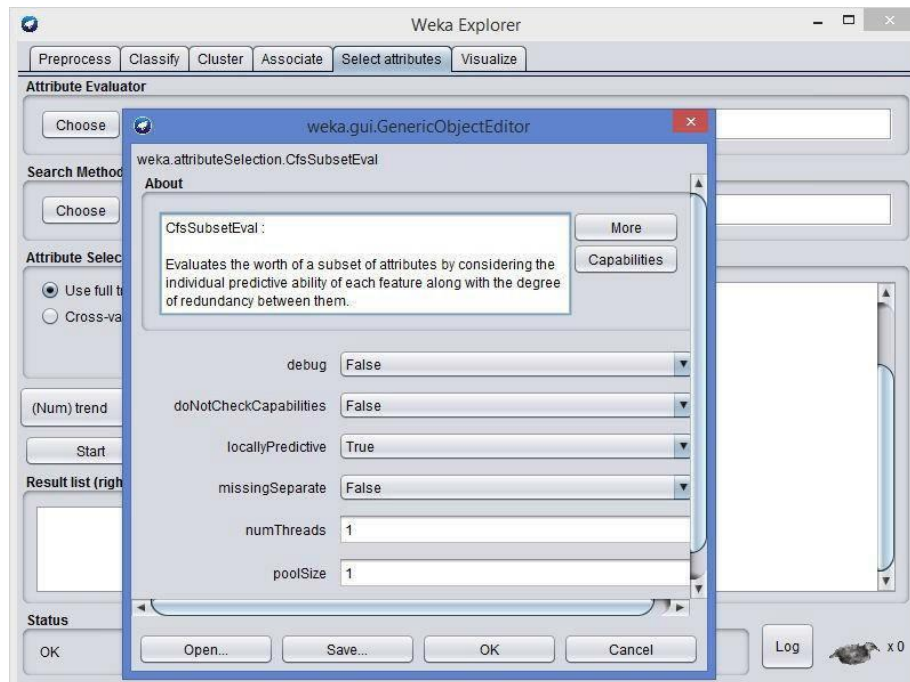


Fig: 1.2 Pre-processor



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

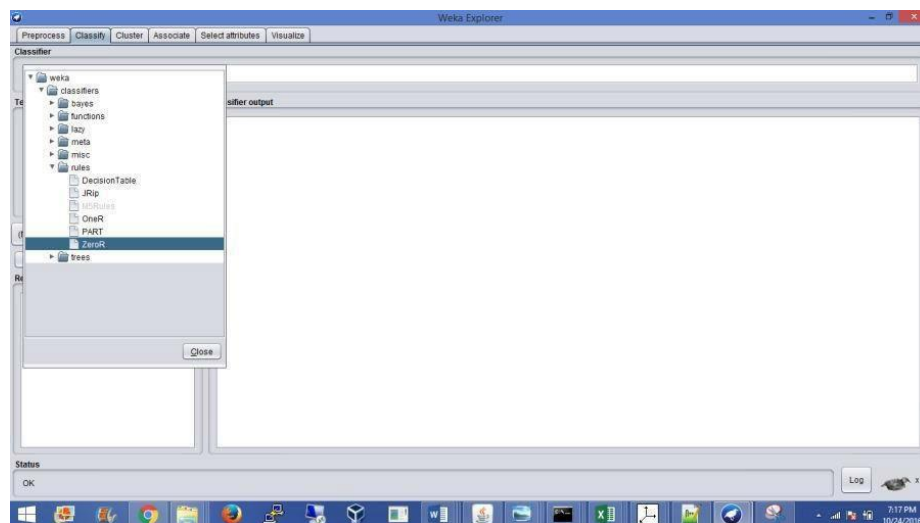


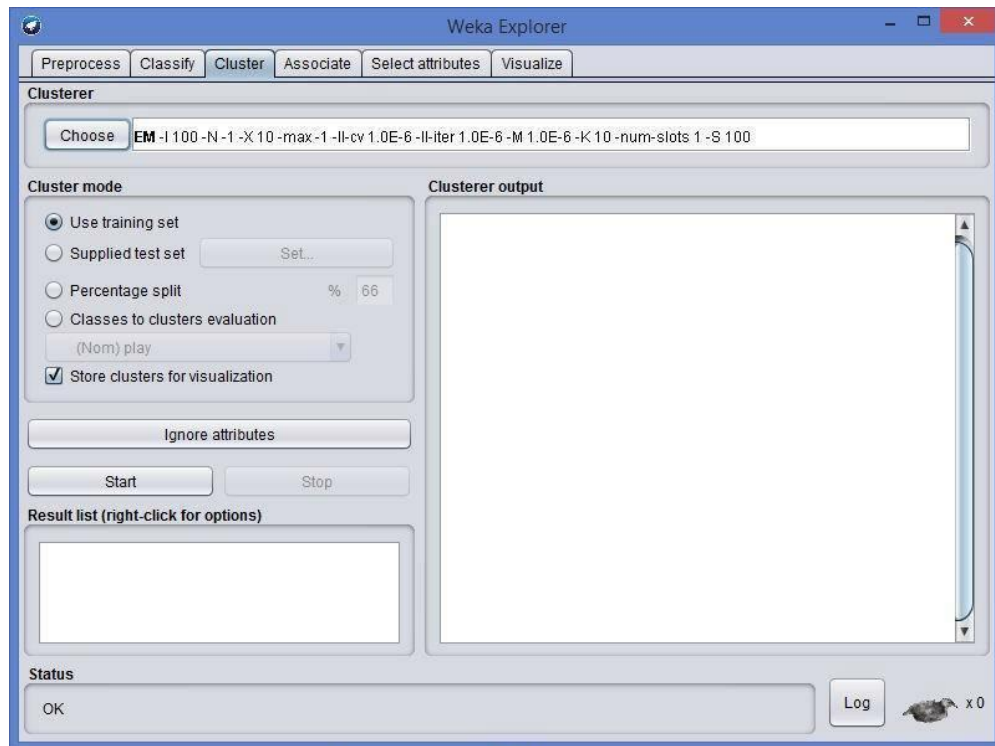
Fig: 1.3 choosing Zero set from classify

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.

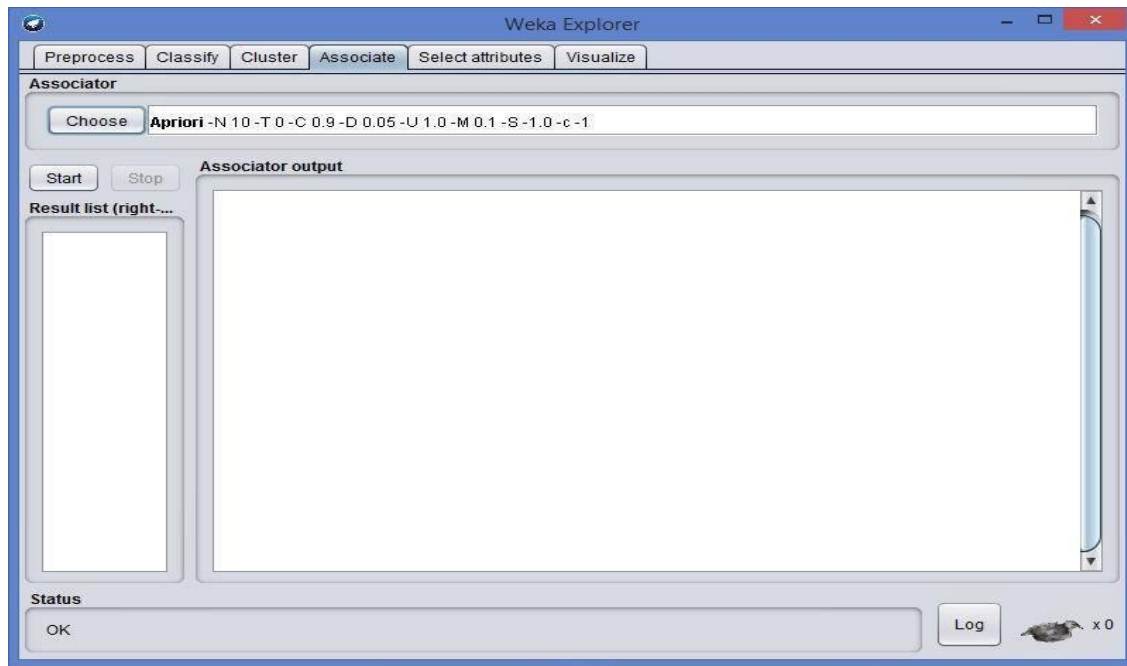
1. Use training set
2. Supplied training set
3. Cross validation
4. Split percentage

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.



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 the dataset.



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.

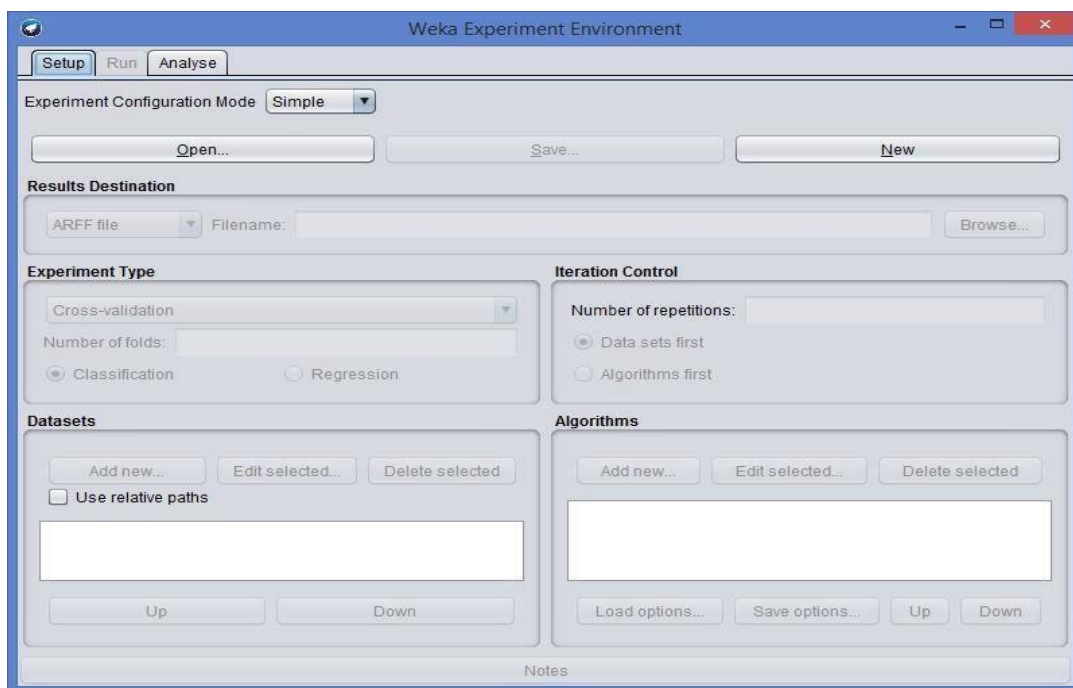


Fig: 1.6 Weka experiment

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.

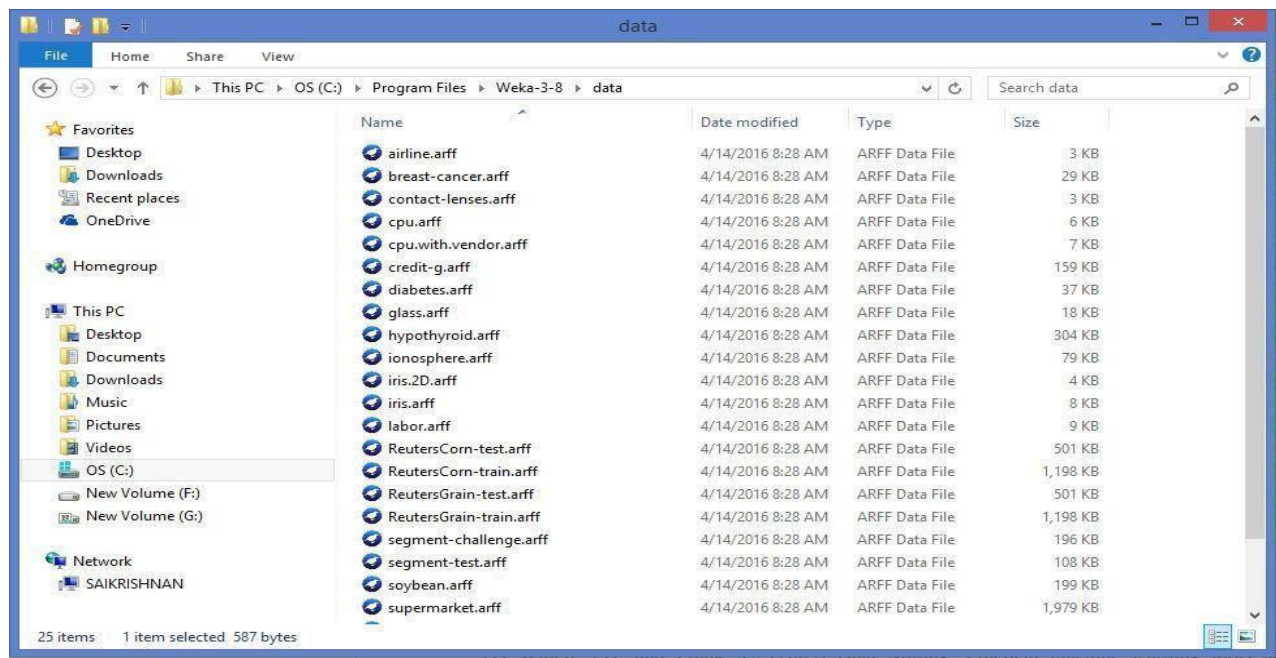


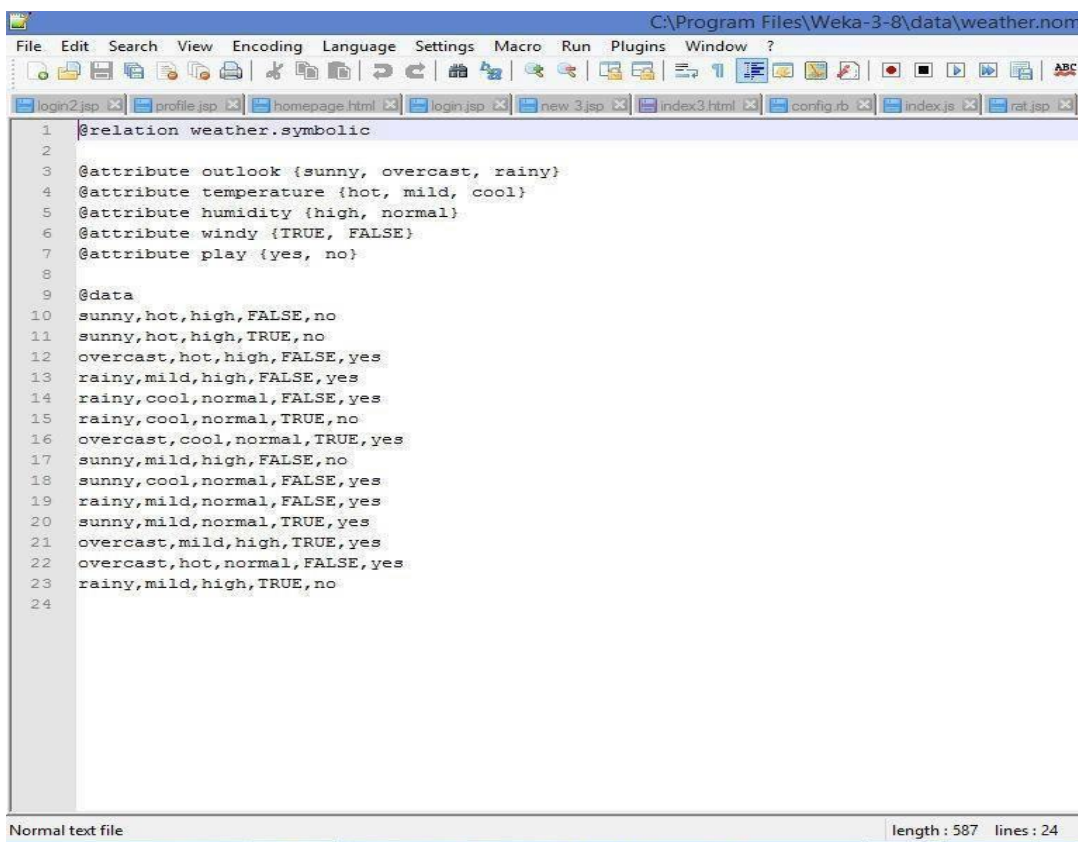
Fig: 1.7 Different Data Sets in weka

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.

A screenshot of a text editor window titled 'C:\Program Files\Weka-3-8\data\weather.nom'. The editor displays an ARFF file for weather data. The header section (lines 1-7) defines the relation 'weather.symbolic' and its attributes: outlook (sunny, overcast, rainy), temperature (hot, mild, cool), humidity (high, normal), windy (TRUE, FALSE), and play (yes, no). The data section (lines 9-23) lists 14 instances of weather data. Line 24 is empty. The status bar at the bottom indicates 'Normal text file' and 'length : 587 lines : 24'.

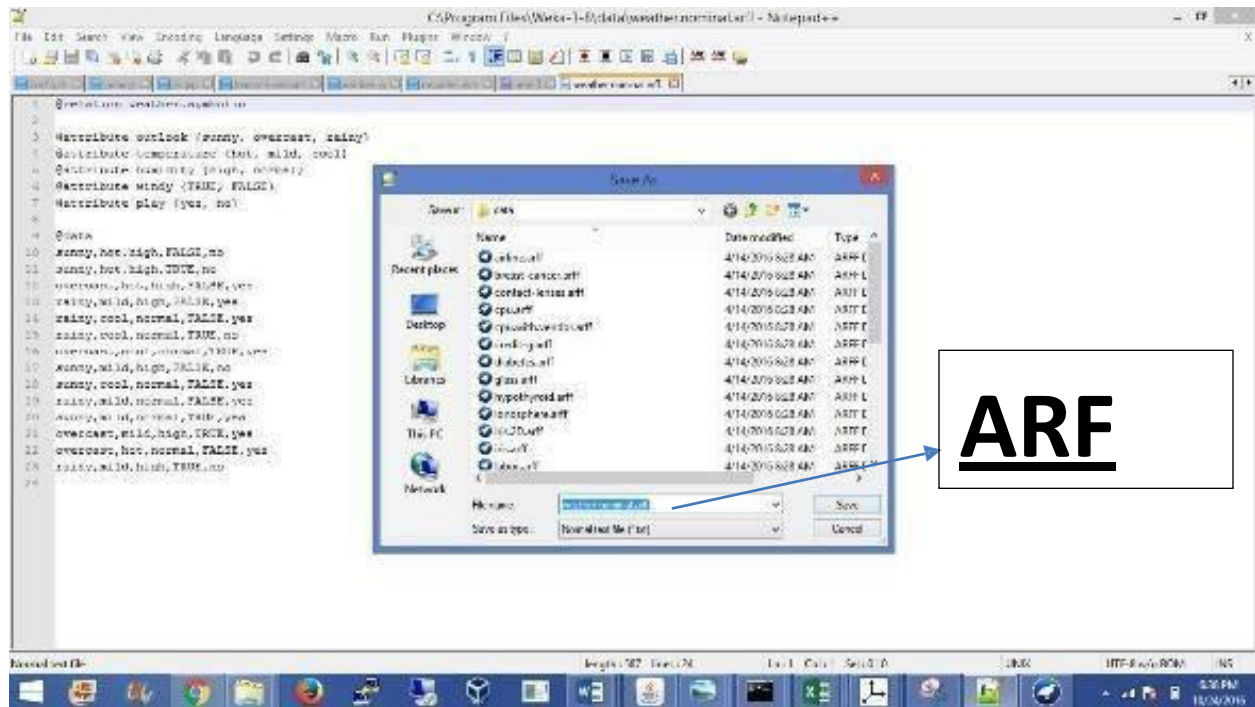
```
1 %relation weather.symbolic
2
3 @attribute outlook {sunny, overcast, rainy}
4 @attribute temperature {hot, mild, cool}
5 @attribute humidity {high, normal}
6 @attribute windy {TRUE, FALSE}
7 @attribute play {yes, no}
8
9 @data
10 sunny,hot,high,FALSE,no
11 sunny,hot,high,TRUE,no
12 overcast,hot,high,FALSE,yes
13 rainy,mild,high,FALSE,yes
14 rainy,cool,normal,FALSE,yes
15 rainy,cool,normal,TRUE,no
16 overcast,cool,normal,TRUE,yes
17 sunny,mild,high,FALSE,no
18 sunny,cool,normal,FALSE,yes
19 rainy,mild,normal,FALSE,yes
20 sunny,mild,normal,TRUE,yes
21 overcast,mild,high,TRUE,yes
22 overcast,hot,normal,FALSE,yes
23 rainy,mild,high,TRUE,no
24
```

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



Exercise:

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

Experiment 3: Pre-Processes Techniques on Data Set

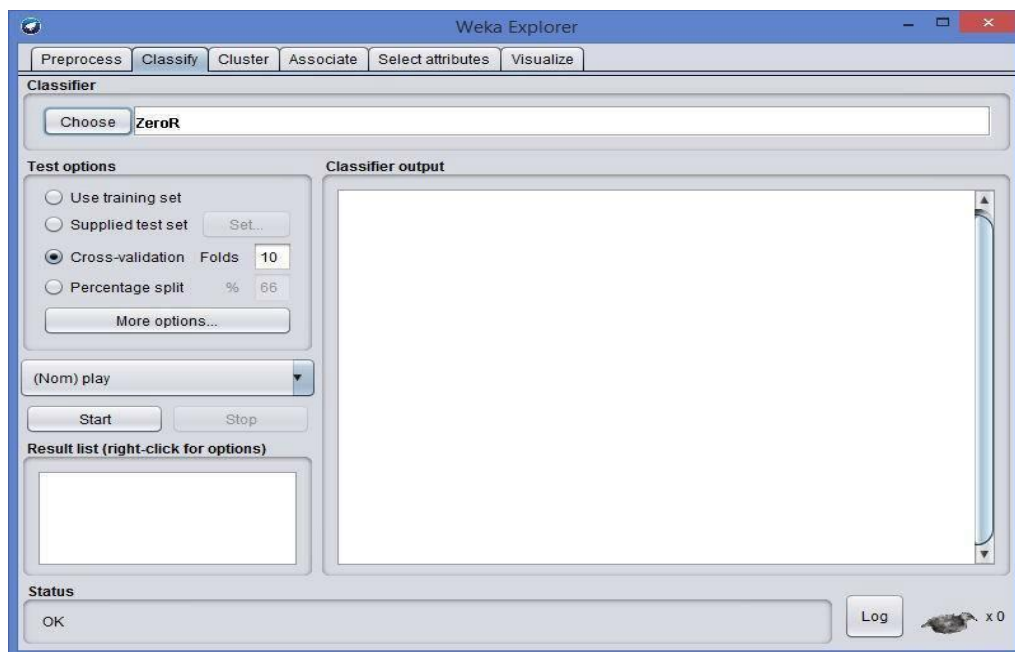
Aim: 3a) Pre-process a given dataset based on Attribute selection

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 to assign a worth to each subset of attributes. The search method will determine what style of search to perform. The options that you can set for selection in the „Attribute Selection Mode“ figno: 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.

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

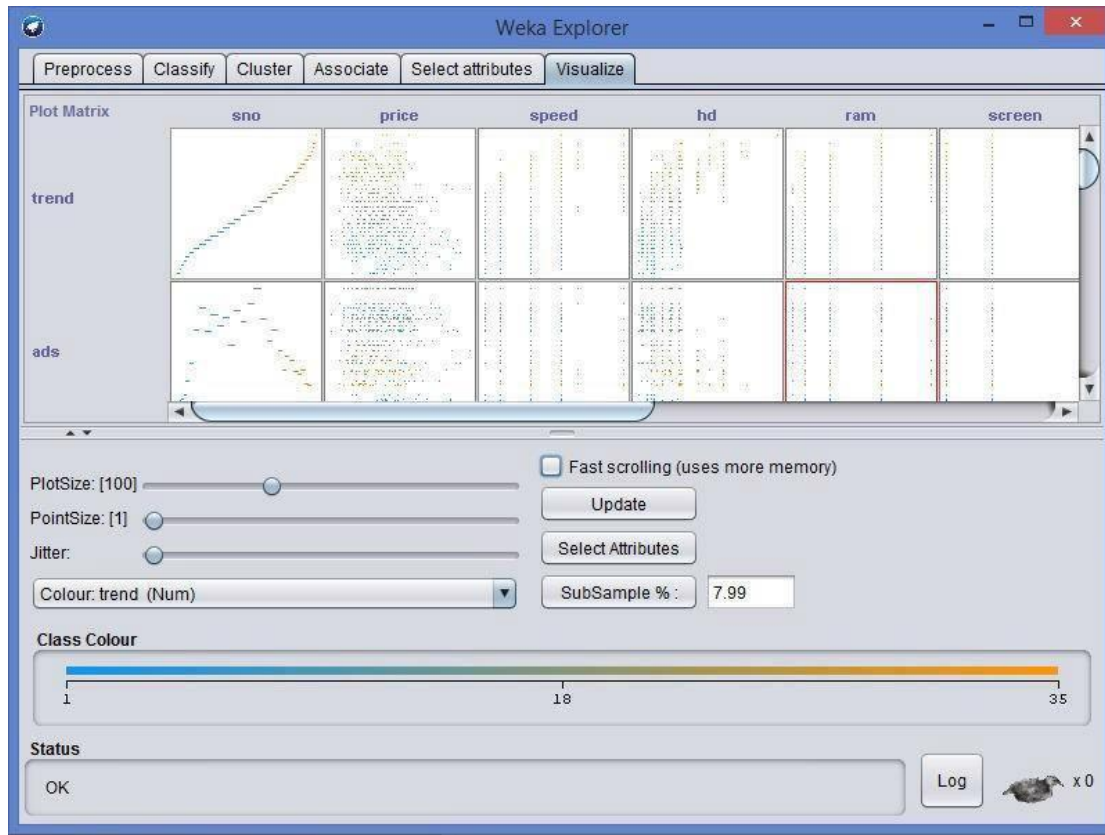


Fig: 3.2 Data Visualization

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.

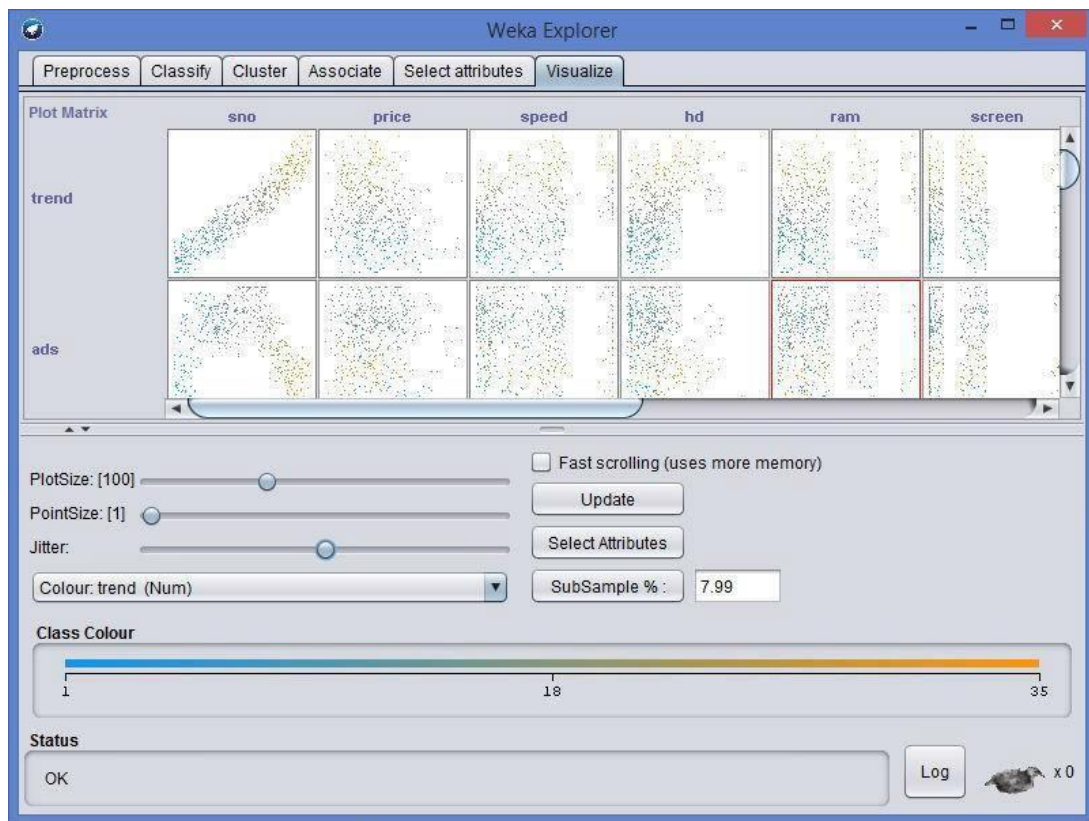


Fig 3.3: Preprocessing with jitter

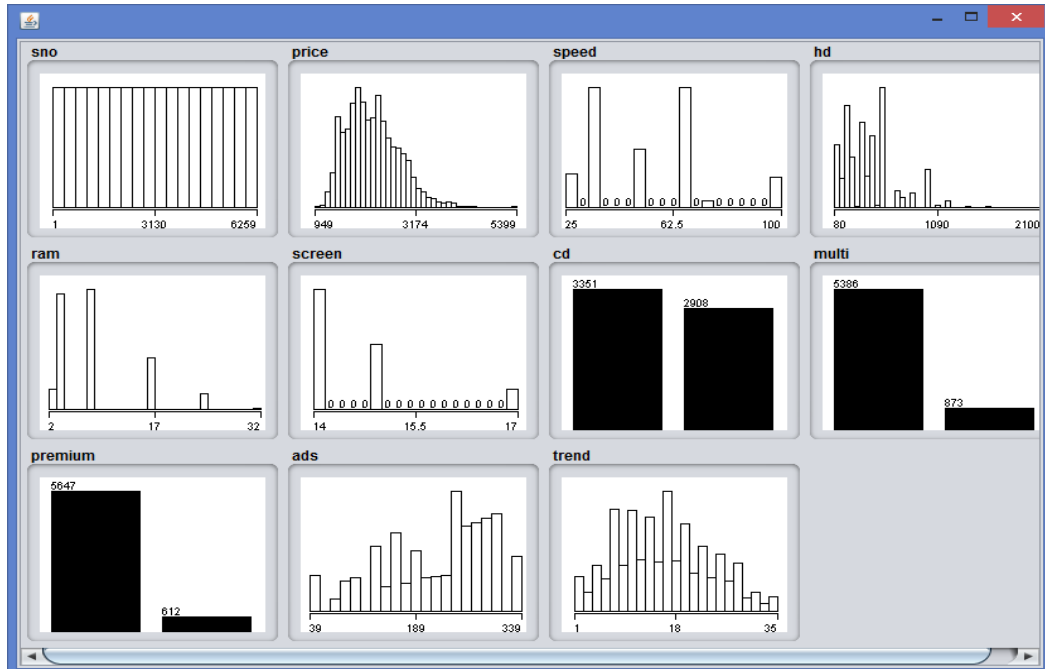


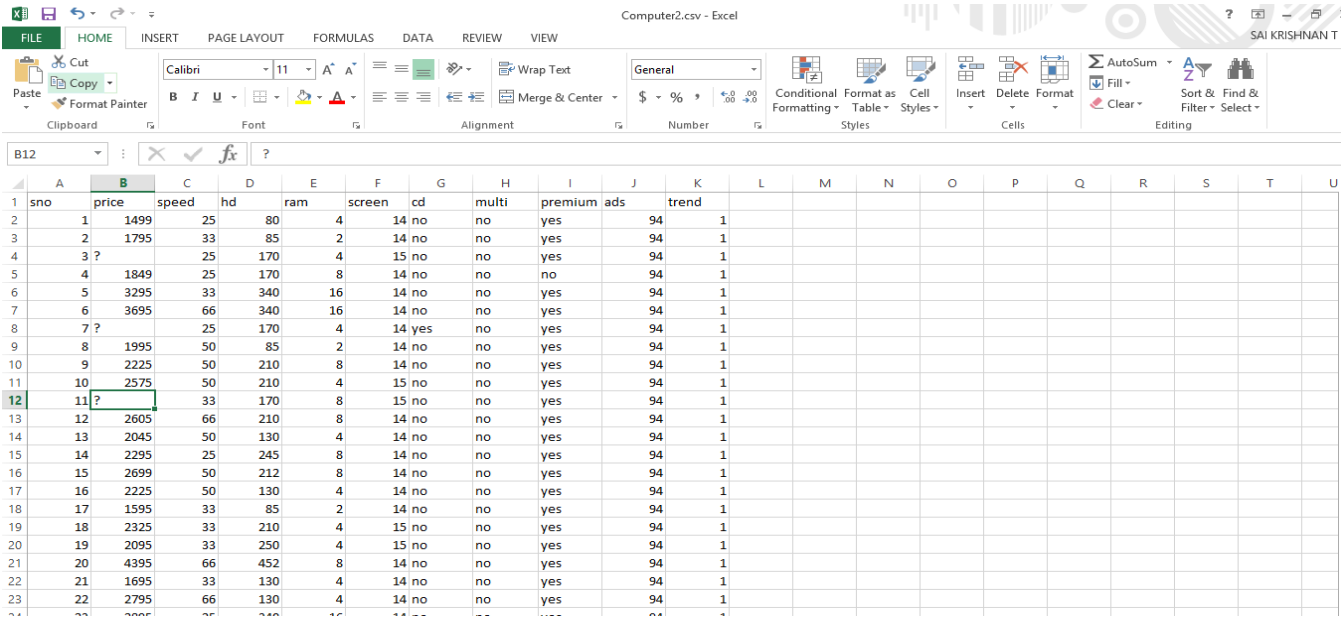
Fig. 3.3 Data visualization

Exercis

e 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



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
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

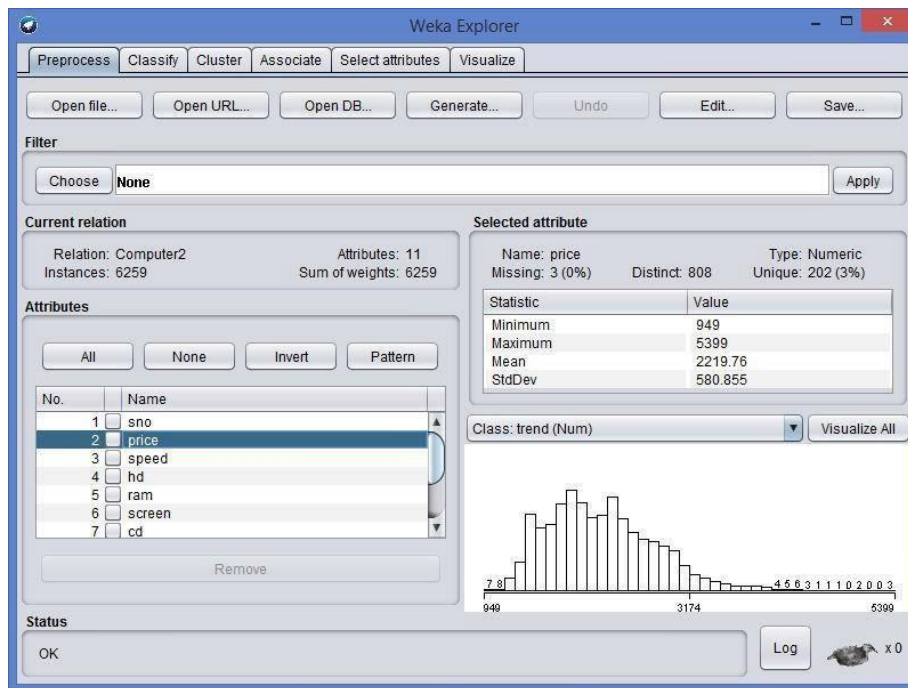
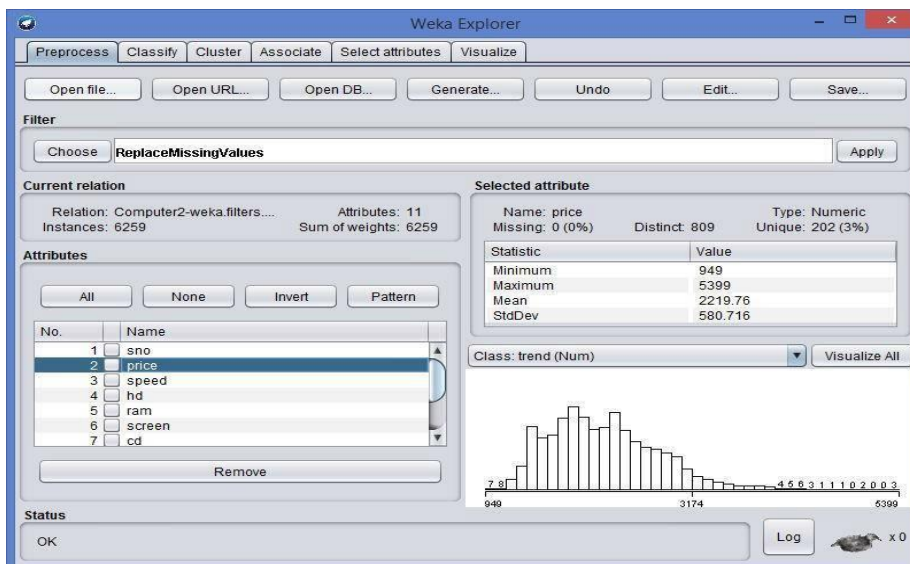
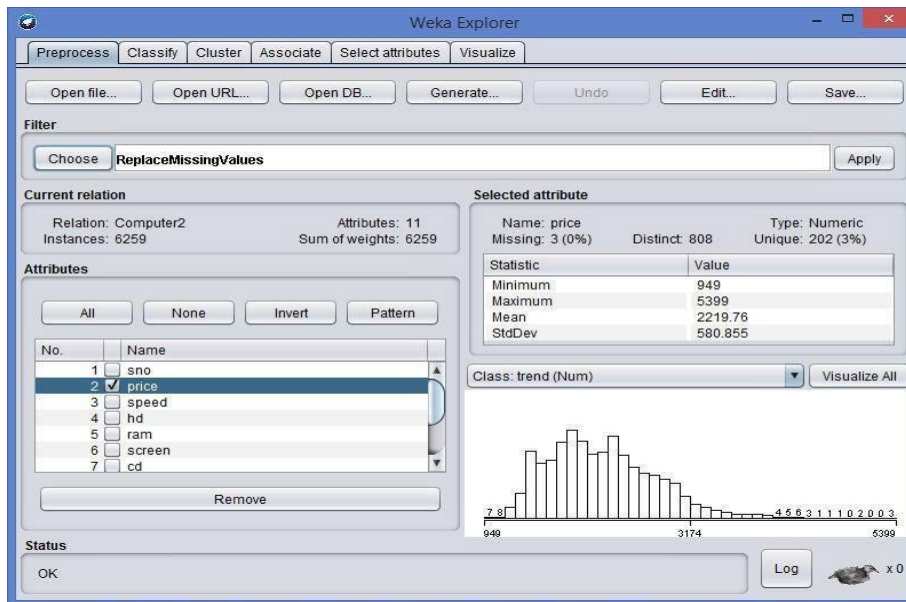
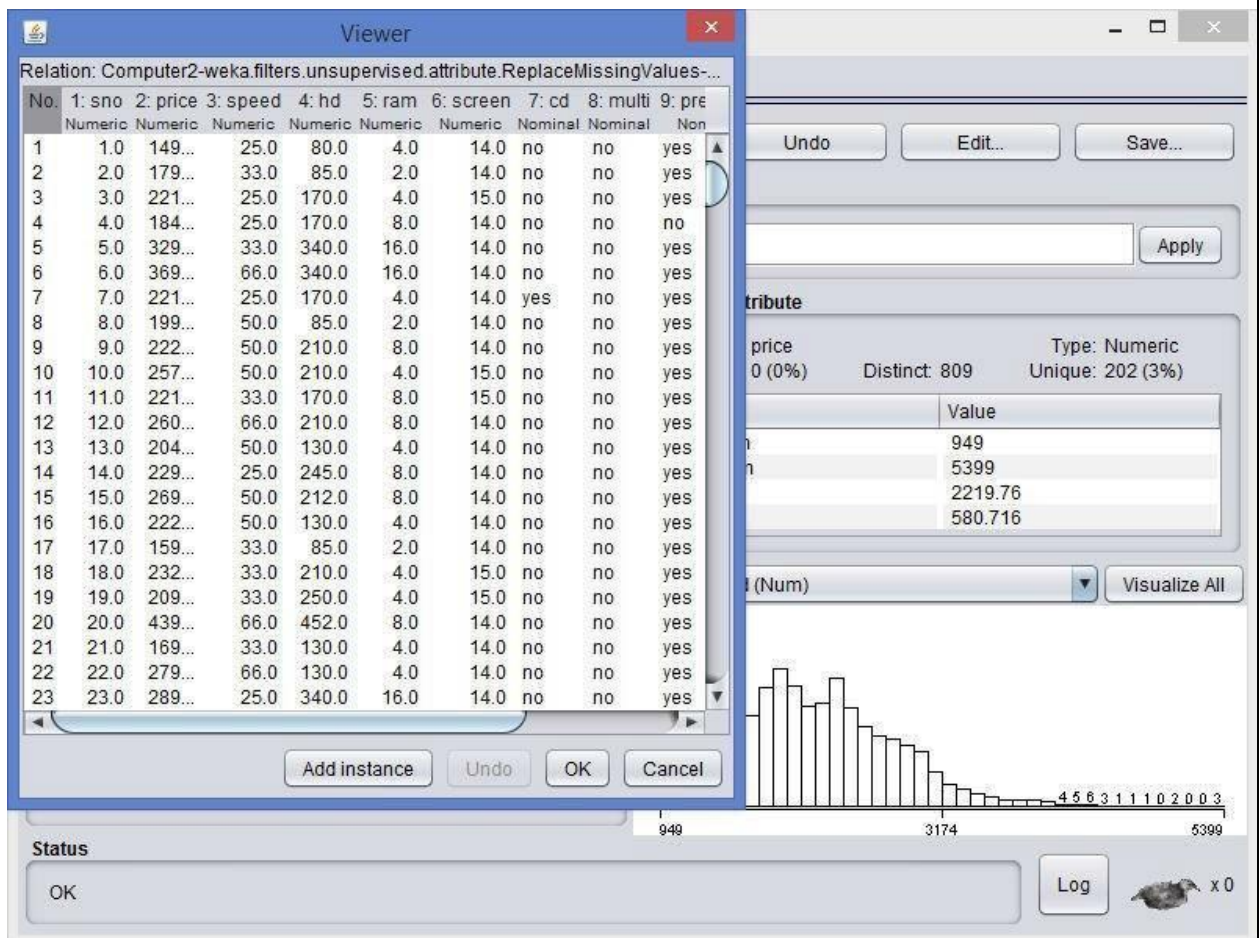


Fig: 3.5 Choosing a dataset

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.





ComputerReplaced.csv - Excel

	sno	price	speed	hd	ram	screen	cd	multi	premium	ads	trend
1	1	1499	25	80	4	14	no	yes	94	1	
2	2	1795	33	85	2	14	no	no	yes	94	1
3	3	2219.76	25	170	4	15	no	yes	94	1	
4	4	1849	25	170	8	14	no	no	yes	94	1
5	5	3295	33	340	16	14	no	yes	94	1	
6	6	3695	66	340	16	14	no	yes	94	1	
7	7	2219.76	25	170	4	14	yes	no	yes	94	1
8	8	1995	50	85	2	14	no	no	yes	94	1
9	9	2225	50	210	8	14	no	no	yes	94	1
10	10	2575	50	210	4	15	no	no	yes	94	1
11	11	2219.76	33	170	8	15	no	no	yes	94	1
12	12	2605	66	210	8	14	no	no	yes	94	1
13	13	2045	50	130	4	14	no	no	yes	94	1
14	14	2295	25	245	8	14	no	no	yes	94	1
15	15	2699	50	212	8	14	no	no	yes	94	1
16	16	2225	50	130	4	14	no	no	yes	94	1
17	17	1595	33	85	2	14	no	no	yes	94	1
18	18	2325	33	210	4	15	no	no	yes	94	1
19	19	2095	33	250	4	15	no	no	yes	94	1
20	20	4395	66	452	8	14	no	no	yes	94	1
21	21	1695	33	130	4	14	no	no	yes	94	1
22	22	2795	66	130	4	14	no	no	yes	94	1
23	23	2895	25	340	16	14	no	no	yes	94	1

Fig: 3.6 Replaced values

Exercise

1. Create your own dataset having missing values included.

Experiment 4 : Demonstration of classification rule on a given dataset

Aim: This experiment illustrates the use of j48 classifier in weka. The sample data set used in this experiment is “student” data available at arff format. This document assumes that appropriate data pre processing has been performed.

Steps involved in this experiment:

Step-1: We begin the experiment by loading the data (student.arff) into weka.

Step2: Next we select the “classify” tab and click “choose” button to select the “j48” classifier.

Step3: Now we specify the various parameters. These can be specified by clicking in the text box to the right of the choose button. In this example, we accept the default values. The default version does perform some pruning but does not perform error pruning.

Step4: Under the “text” options in the main panel. We select the 10-fold cross validation as our evaluation approach. Since we don’t have separate evaluation data set, this is necessary to get a reasonable idea of accuracy of generated model.

Step-5: We now click “start” to generate the model. The Ascii version of the tree as well as evaluation statistic will appear in the right panel when the model construction is complete.

Step-6: Note that the classification accuracy of model is about 69%. This indicates that we may find more work. (Either in preprocessing or in selecting current parameters for the classification)

Step-7: Now weka also lets us view a graphical version of the classification tree. This can be done by right clicking the last result set and selecting “visualize tree” from the pop-up menu.

Step-8: We will use our model to classify the new instances.

Step-9: In the main panel under “text” options click the “supplied test set” radio button and then click the “set” button. This will pop-up a window which will allow you to open the file containing test instances.

Dataset student .arff

@relation student

@attribute age {<30,30-40,>40}

@attribute income {low, medium,
high}@attribute student {yes, no}

@attribute credit-rating {fair,
excellent}@attribute buyspc {yes, no}

@data

%

<30, high, no, fair, no

<30, high, no, excellent,

no30-40, high, no, fair,

yes

>40, medium, no, fair, yes

>40, low, yes, fair, yes

>40, low, yes, excellent, no

30-40, low, yes, excellent,

yes

<30, medium, no, fair, no

<30, low, yes, fair, no

>40, medium, yes, fair, yes

<30, medium, yes, excellent, yes

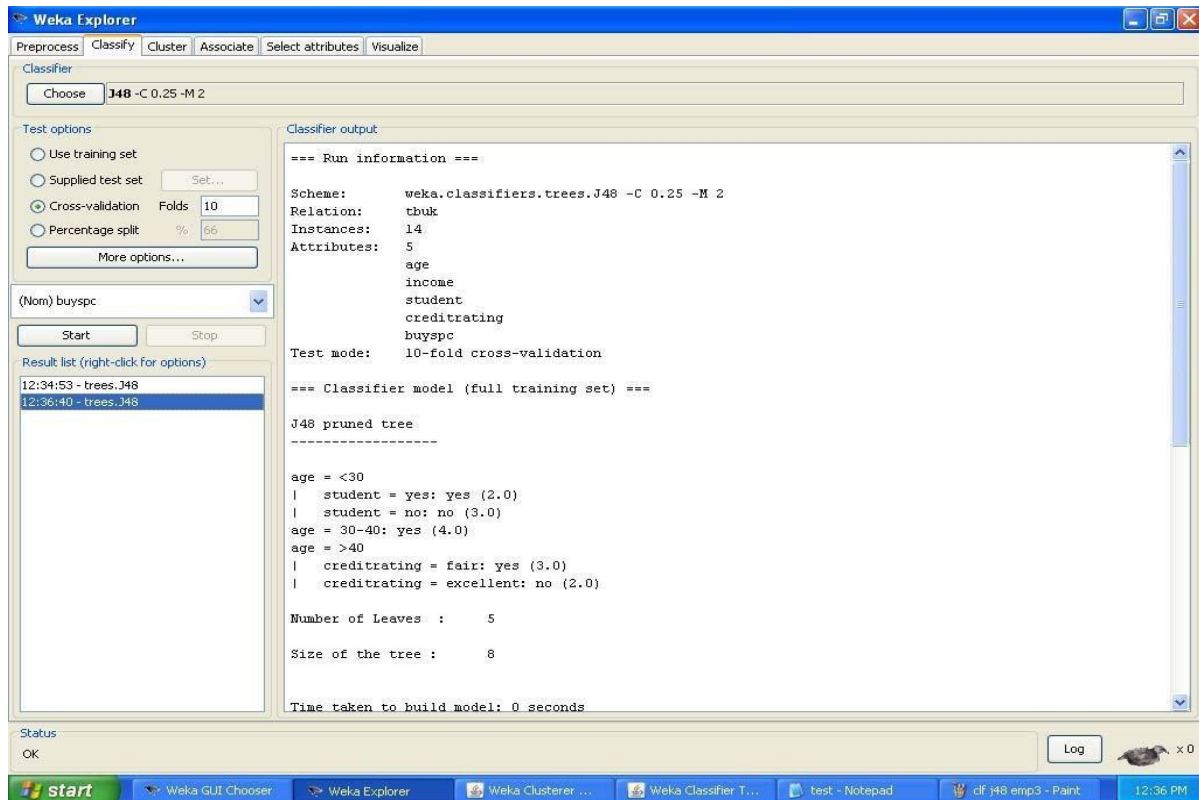
30-40, medium, no, excellent,

yes30-40, high, yes, fair, yes

>40, medium, no, excellent, no

%

The following screenshot shows the classification rules that were generated when j48algorithm is applied on the given dataset.



Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

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) buyspc

Start Stop

Result list (right-click for options)

12:34:53 - trees.J48

12:36:40 - trees.J48

Classifier output

Size of the tree : 8

Time taken to build model: 0 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	ROC Area	Class
	0.556	0.6	0.625	0.556	0.588	0.633	yes
	0.4	0.444	0.333	0.4	0.364	0.633	no
Weighted Avg.	0.5	0.544	0.521	0.5	0.508	0.633	

=== Confusion Matrix ===

a b <-- classified as

5	4	a = yes
3	2	b = no

Status

OK

Log x 0

start Weka GUI Chooser Weka Explorer Weka Clusterer ... Weka Classifier T... test - Notepad cf j48 stud1 - Paint 12:37 PM

Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **J48 -C 0.25 -M 2**

Test options

- ☐ Use training set
- ☐ Supplied test set **Set...**
- ☒ Cross-validation **Test on a user-specified dataset**
- ☐ Percentage split % **66**

More options...

(Nom) buyspc

Start **Stop**

Result list (right-click for options)

- 12:34:53 - trees.J48
- 12:36:40 - trees.J48

Classifier output

Size of the tree : 8

Weka Classifier Tree Visualizer: 12:36:40 - trees.J48 (tbuk)

Tree View

```
graph TD; age -- "<=30" --> student; age -- "= 30-40" --> yes40["yes (4.0)"]; age -- ">40" --> creditrating; student -- "= yes" --> yes20["yes (2.0)"]; student -- "= no" --> no30["no (3.0)"]; creditrating -- "= fair" --> yes30["yes (3.0)"]; creditrating -- "= excellent" --> no20["no (2.0)"];
```

Class

yes

no

Status

OK

Log

start | Weka GUI C... | Weka Explorer | Weka Cluste... | Weka Classifi... | Weka Classifi... | test - Notepad | clf j48 stud2 ... | 12:37 PM

Experiment 5 : Generate Association Rules using the 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's 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.

$\{2\} \Rightarrow \{3,5\}$: (If „2“ is bought , what“s the probability that „3“ and „5“ would be bought in same transaction)

$$\text{Confidence} = P(\{3,5\}|\{2\}) / P(\{2\}) = 2/3 = 67\%$$

$$\{3\} \Rightarrow \{2,5\} = P(\{3,5\}|\{3\}) = 2/3 = 67\%$$

$$\{5\} \Rightarrow \{2,3\} = P(\{3,5\}|\{5\}) = 2/3 = 67\%$$

$$\{2,3\} \Rightarrow \{5\} = P(\{3,5\}|\{2,3\}) = 2/2 = 100\%$$

$$\{3,5\} \Rightarrow \{2\} = P(\{3,5\}|\{3,5\}) = 2/2 = 100\%$$

$$\{2,5\} \Rightarrow \{3\} = P(\{3,5\}|\{2,5\}) = 2/3 = 67\%$$

Also, considering the remaining 2-items sets, we would get the following associations-

$$\{1\} \Rightarrow \{3\} = P(\{1,3\}|\{1\}) = 2/2 = 100\%$$

$$\{3\} \Rightarrow \{1\} = P(\{1,3\}|\{3\}) = 2/3 = 67\%$$

$$\{2\} \Rightarrow \{3\} = P(\{3,2\}|\{2\}) = 2/3 = 67\%$$

$$\{3\} \Rightarrow \{2\} = P(\{3,2\}|\{3\}) = 2/3 = 67\%$$

$$\{2\} \Rightarrow \{5\} = P(\{2,5\}|\{2\}) = 3/3 = 100\%$$

$$\{5\} \Rightarrow \{2\} = P(\{2,5\}|\{5\}) = 3/3 = 100\%$$



$$\{3\} \Rightarrow \{5\} = P(\{3,5\}|\{3\}) = 2/3 = 67\%$$

$$\{5\} \Rightarrow \{3\} = P(\{3,5\}|\{5\}) = 2/3 = 67\%$$

Eliminate all those having confidence less than 70%. Hence, the rules would be –

$\{2,3\} \Rightarrow \{5\}$, $\{3,5\} \Rightarrow \{2\}$, $\{1\} \Rightarrow \{3\}$, $\{2\} \Rightarrow \{5\}$, $\{5\} \Rightarrow \{2\}$.

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

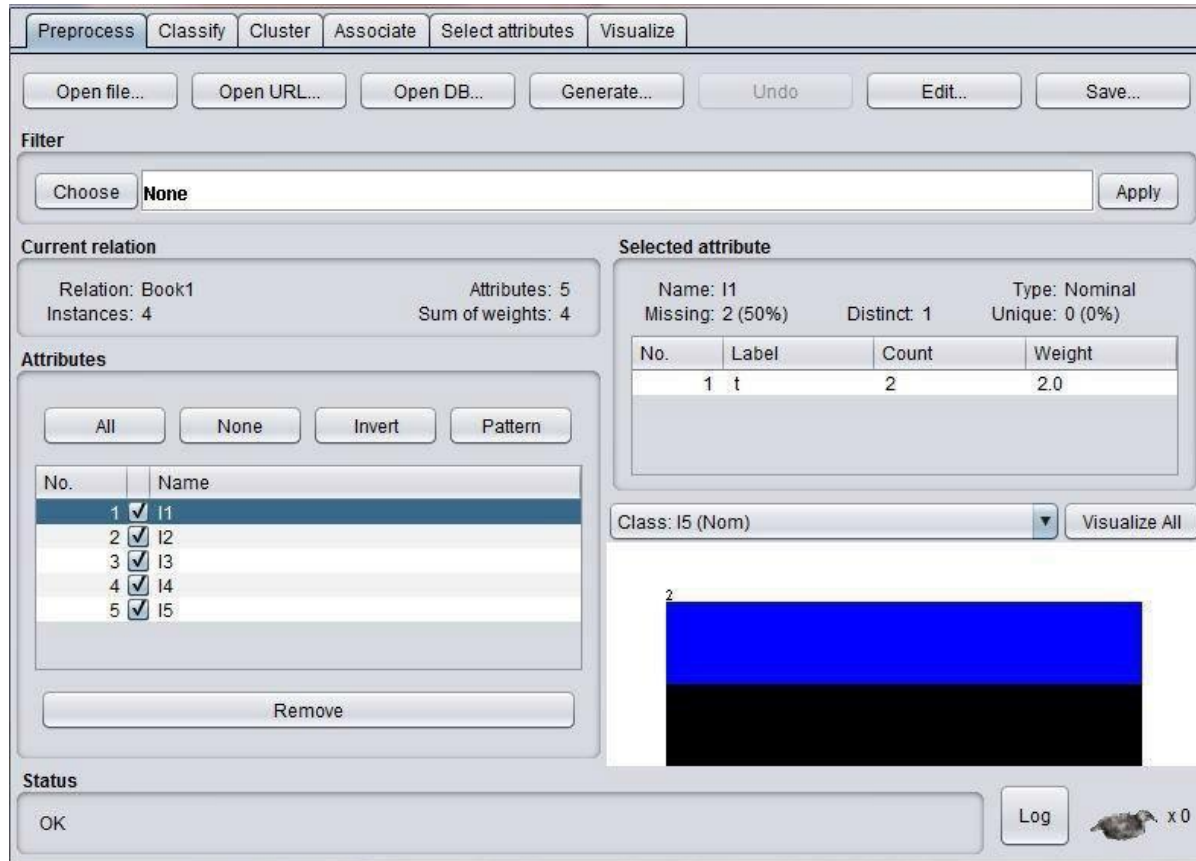
Clipboard		Font				
A1			 I1			
	A	B	C	D	E	F
1	I1	I2	I3	I4	I5	
2	t		t	t		
3		t	t		t	
4	t	t	t		t	
5		t			t	
6						
7						
8						
9						

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

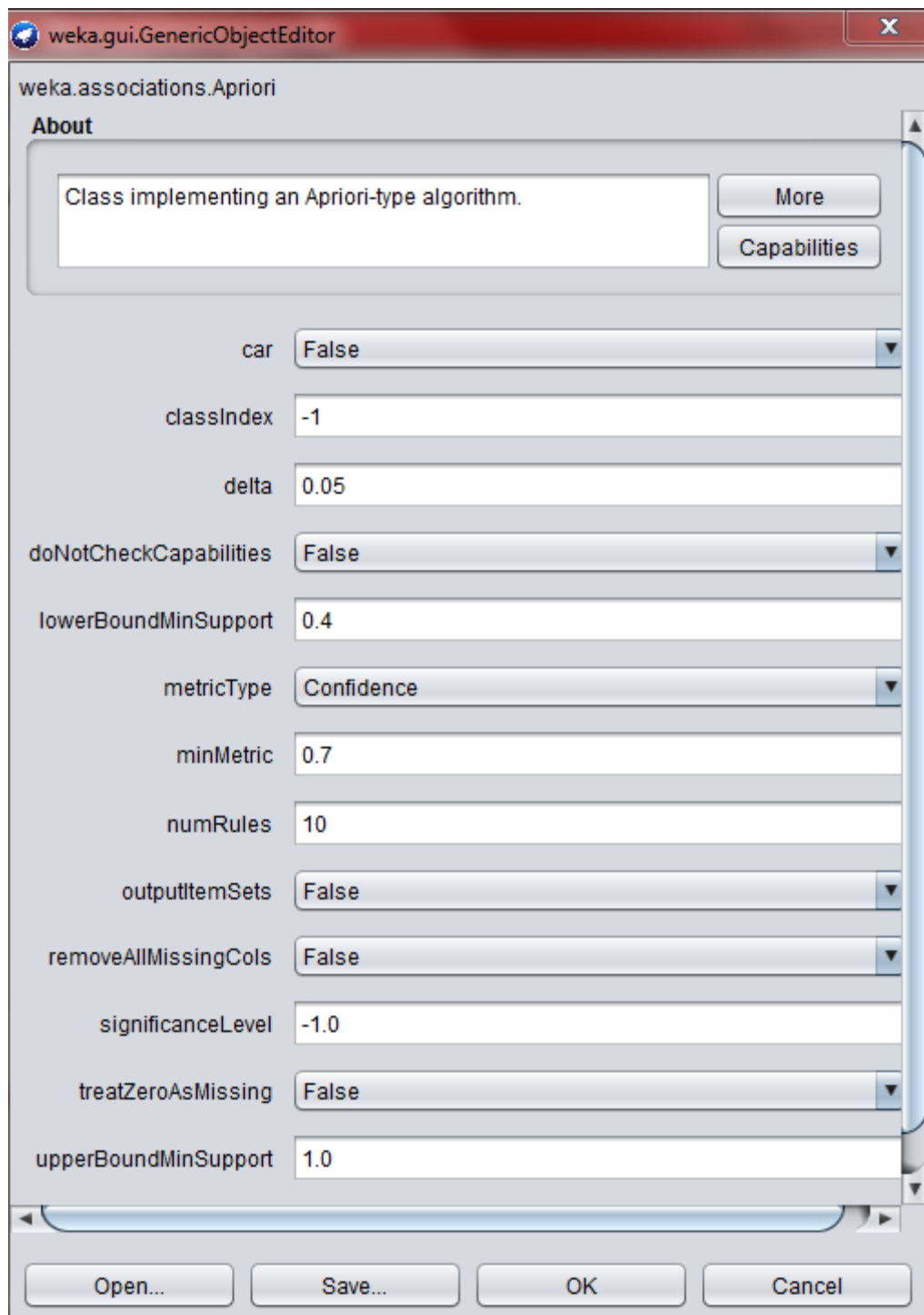
Procedure for running the rules in weka:

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.



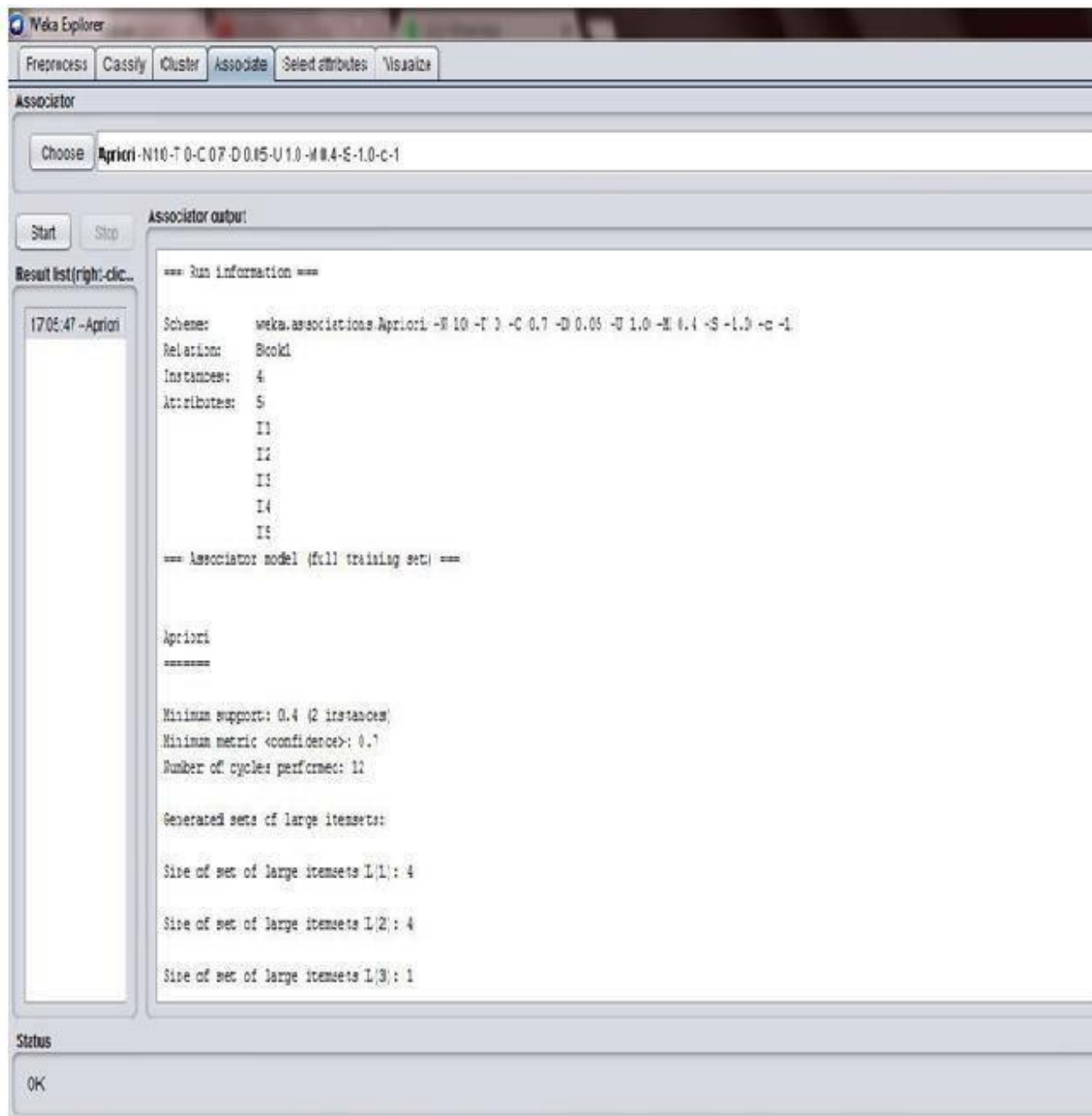
Step 2: Now select the association tab and then choose apriori algorithm by setting the minimum support and confidence as shown in the figure



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:



Associator output

```
14
15
=== 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 parkinsons and neurological diseases using voice recording dataset.
[<https://archive.ics.uci.edu/ml/machine-learning-databases/00000/> refer this link for datasets]

Experiment 6 : Generating Association Rules Using 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.

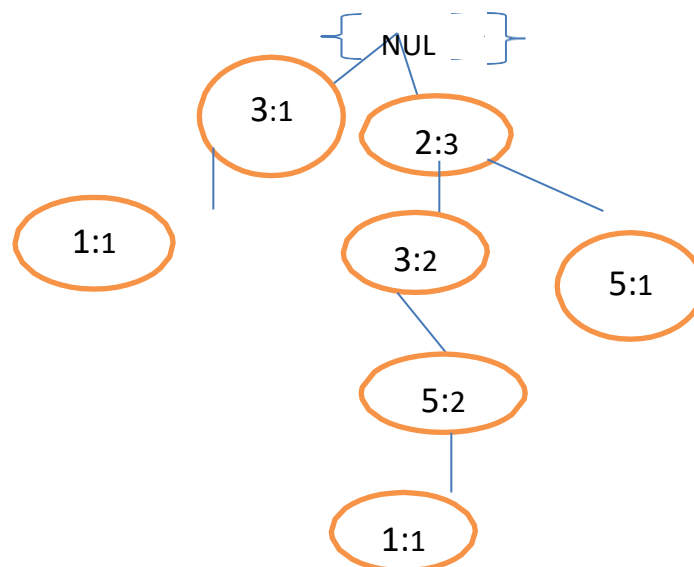
Prefixes:

- 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-

- $\{3\} \Rightarrow \{1\} = 2/3 = 67\%$
- $\{1\} \Rightarrow \{3\} = 2/2 = 100\%$
- $\{2\} \Rightarrow \{3,5\} = 2/3 = 67\%$

- $\{2,5\} \Rightarrow \{3\} = 2/3 = 67\%$
- $\{3,5\} \Rightarrow \{2\} = 2/2 = 100\%$
- $\{2,3\} \Rightarrow \{5\} = 2/2 = 100\%$
- $\{3\} \Rightarrow \{2,5\} = 2/3 = 67\%$
- $\{5\} \Rightarrow \{2,3\} = 2/3 = 67\%$
- $\{2\} \Rightarrow \{5\} = 3/3 = 100\%$
- $\{5\} \Rightarrow \{2\} = 3/3 = 100\%$
- $\{2\} \Rightarrow \{3\} = 2/3 = 67\%$
- $\{3\} \Rightarrow \{2\} = 2/3 = 67\%$

Thus eliminating all the sets having confidence less than 70%, we obtain the following conclusions:

$\{1\} \Rightarrow \{3\}$, $\{3,5\} \Rightarrow \{2\}$, $\{2,3\} \Rightarrow \{5\}$, $\{2\} \Rightarrow \{5\}$, $\{5\} \Rightarrow \{2\}$.

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

Clipboard		Font				
A1		f _x				
		I1				
	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.

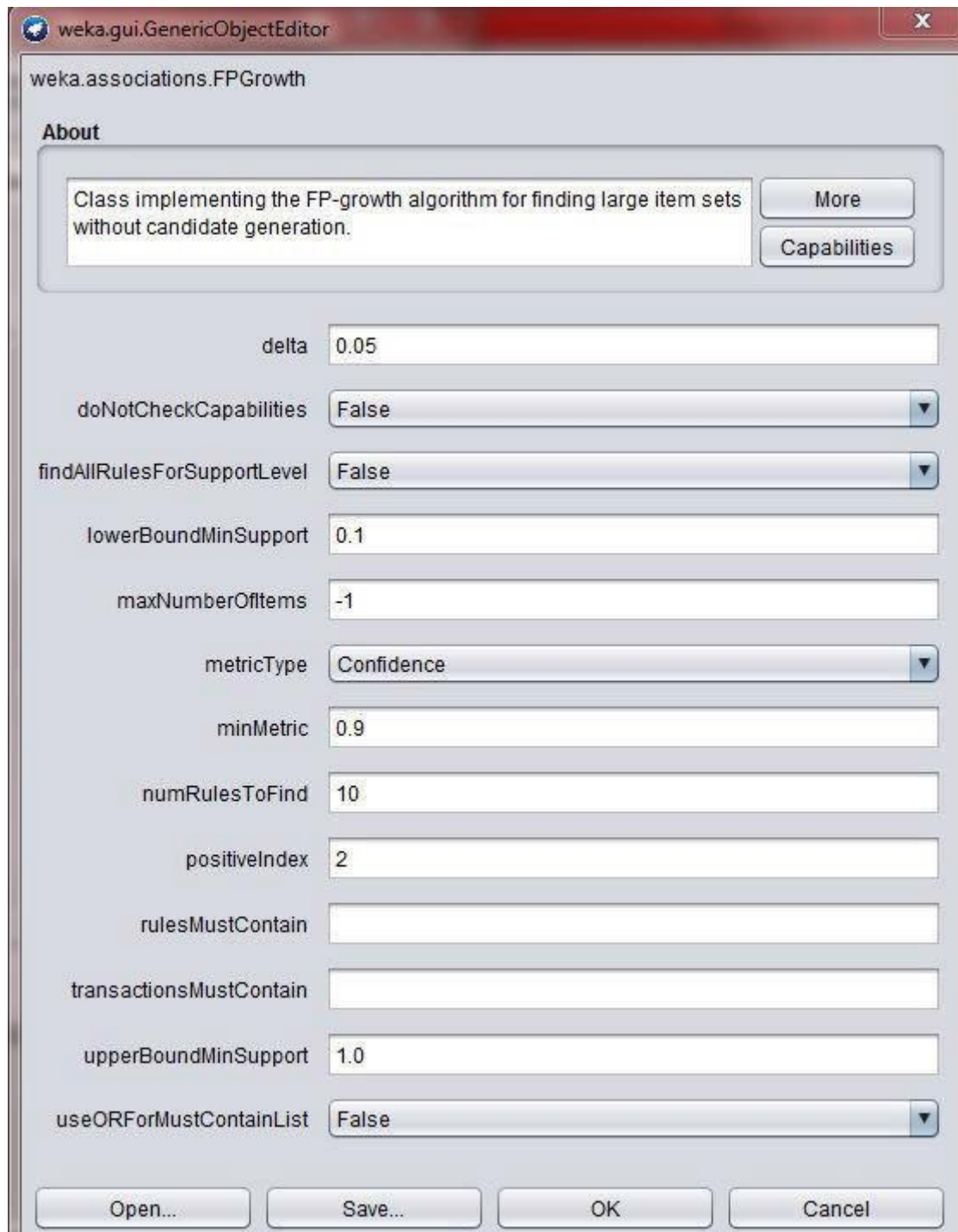
Procedure for running the rules in weka:

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.



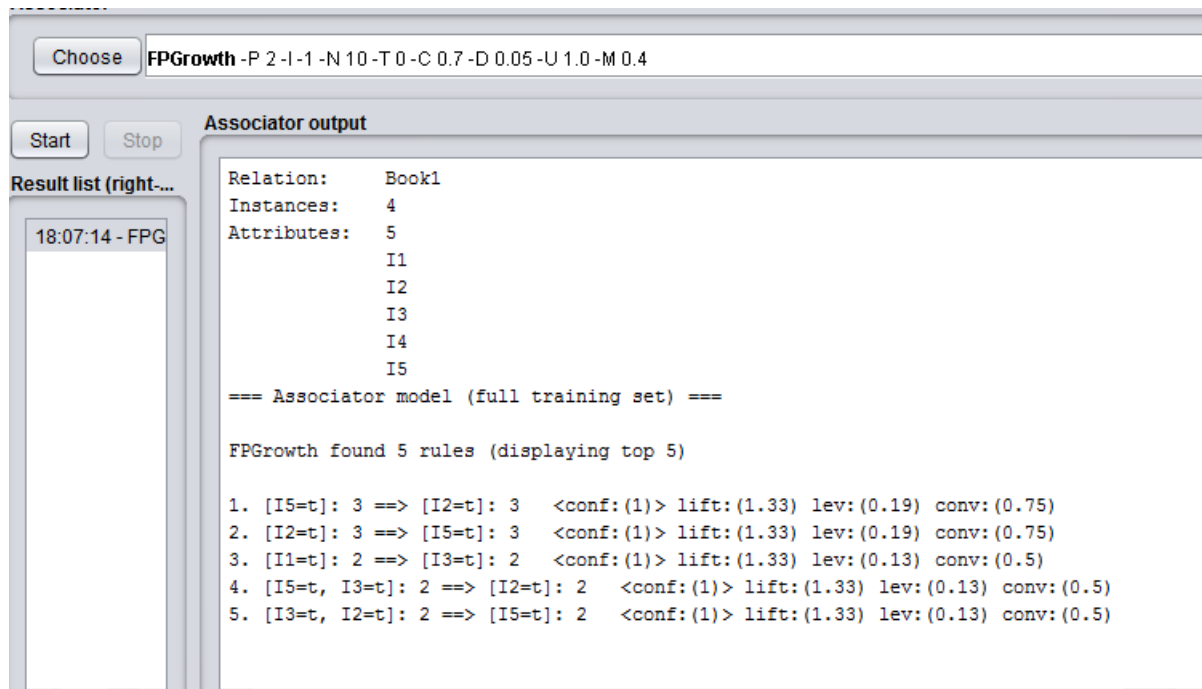
Step 2: Now select the association tab and then choose FPgrowth algorithm by setting the minimum support and confidence as shown in the figure.



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:



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

Experiment 7: Naïve bayes classification on a given data set

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	Medium	No	Fair	Yes
>40	Low	Yes	Fair	Yes
>40	Low	Yes	Excellent	No
31-40	Medium	Yes	Excellent	Yes
<=30	Low	No	Fair	No
<=30	Medium	Yes	Fair	Yes
>40	Medium	Yes	Fair	Yes
<30	Medium	Yes	Excellent	Yes
31-40	Medium	No	Excellent	Yes
31-40	High	Yes	Fair	Yes
>40	Medium	No	Excellent	No

CLASS:

C1:buys_com
puter = 'yes'

C2:buys_com

puter='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 $p(x/c2)$ we get:

1. $P(\text{age} \leq 30 \mid \text{buys_computer} = \text{"yes"}) = 2/9$
2. $P(\text{age} \leq 30 \mid \text{buys_computer} = \text{"no"}) = 3/5$
3. $P(\text{income} = \text{"medium"} \mid \text{buys_computer} = \text{"yes"}) = 4/9$
4. $P(\text{income} = \text{"medium"} \mid \text{buys_computer} = \text{"no"}) = 2/5$
5. $P(\text{student} = \text{"yes"} \mid \text{buys_computer} = \text{"yes"}) = 6/9$
6. $P(\text{student} = \text{"yes"} \mid \text{buys_computer} = \text{"no"}) = 1/5 = 0.2$
7. $P(\text{credit_rating} = \text{"fair"} \mid \text{buys_computer} = \text{"yes"}) = 6/9$
8. $P(\text{credit_rating} = \text{"fair"} \mid \text{buys_computer} = \text{"no"}) = 2/5$

- $X = (\text{age} \leq 30, \text{income} = \text{medium}, \text{student} = \text{yes}, \text{credit_rating} = \text{fair})$
 $P(X/C1): P(X/\text{buys_computer} = \text{"yes"}) = 2/9 * 4/9 * 6/9 * 6/9 = 32/1134$

$$P(X/C2): P(X/\text{buys_computer} = \text{"no"}) = 3/5 * 2/5 * 1/5 * 2/5 = 12/125$$

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

$$P(X/\text{buys_computer} = \text{"yes"}) * P(\text{buys_computer} = \text{"yes"}) = (32/1134) * (9/14) = 0.019$$

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

$$P(X/\text{buys_computer} = \text{"no"}) * P(\text{buys_computer} = \text{"no"}) = (12/125) * (5/14) = 0.007$$

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 arff file will look as shown below:

```

store.arff - Notepad
File Edit Format View Help
@relation store

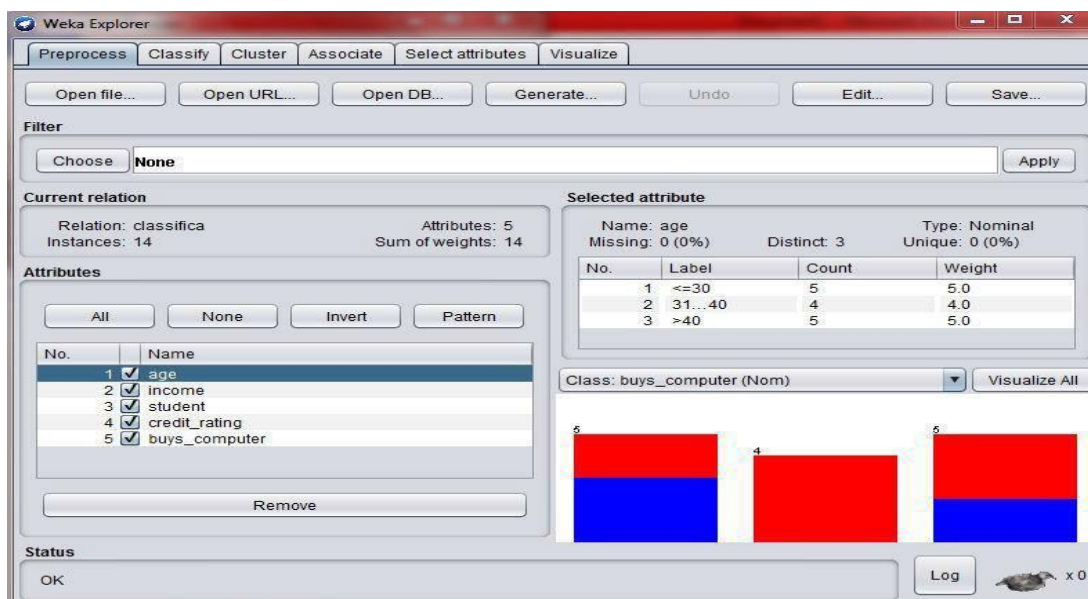
@attribute age {young, middle-aged, old}
@attribute income {high,medium,low}
@attribute student {yes,no}
@attribute credit-rating {fair,excellent}
@attribute buys-computer {yes,no}

@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.



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.

```

Classifier output
=== 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 .

Experiment 8 : Applying k-means clustering on a given data set

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 centroidpoints. 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 shownbelow:

Clipboard		Font		
A1		f_x	i	
	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

Choose **None** Apply

Current relation

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

Attributes

All None Invert Pattern

No.	Name
1	i
2	x1
3	x2

Remove

Selected attribute

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

Class: x2 (Num) Visualize All

Status

OK Log x 0

Step 3:

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

```
Clusterer output

=== 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: 3.22962962963

Initial starting points (random):

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

Missing values globally replaced with mean/mode
```

Final cluster centroids:

Attribute	Cluster#		
	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.