DATA WAREHOUSING AND DATA MINING LABORATORY

(**Semester -VI** of B.Tech)

As per the curricullam and syllabus
of
Bharath Institute of Higher Education & Research

(DWDM Lab Manual)

Bharath
INSTITUTE OF HIGHER EDUCATION AND RESEARCH
(Declared as Deemed - to - be - University under section 3 of UGC Act 1956)
ACCREDITED WITH 'A' GRADE BY NAAC

NEW EDITION

PREPARED BY

DR. M.K.VIDHYALAKSHMI

Bharath

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

(Declared as Deemed-to-be University under section 3 of UGC Act, 1956)

(Vide Notification No. F.9-5/2000 - U.3, Ministry of Human Resource Development, Govt. of India, dated 4th July 2002)

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

LAB MANUAL

SUBJECT NAME: Data Warehousing and Data Mining Laboratory

SUBJECT CODE: BCS6L1

Regulation R2015
(2015-2016)

BCS6L1	DATA WAREHOUSING AND DATA MINING LABORATORY	L	T	P	C
	Total Contact Hours - 30	0	0	3	2
	Prerequisite –Data ware Housing and Data mining				
	Lab Manual Designed by – Dept. of Computer Science and Engineering.				

OBJECTIVES

Data mining is primarily used by the companies with a strong consumer focus. It enables these companies to determine the factors such as price, product positioning, or staff skills, and economic indicators, competition, and customer demographics

COURSE OUTCOMES (COs)

CO1	Provide efficient distribution of information and easy access to data
CO2	Create user friendly reporting environment.
CO3	Find the unseen pattern in large volume of historical data that helps to manage an organization efficiently.
CO4	Understand the concepts of various data mining Techniques.
CO5	Understand the concepts of Preprocessing.
CO6	Explain the concept of Data mining.

MAPPING BETWEEN COURSE OUTCOMES & PROGRAM OUTCOMES
(3/2/1 INDICATES STRENGTH OF CORRELATION) 3- High, 2- Medium, 1-Low

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	3		3							3		2	
CO2	2		3			2						3	2	2	
CO3			3	2	3				3		3	3		2	
CO4	2	2	3		3	2						3		2	
CO5					3							2		2	
CO6					2				2		3	3		2	

Category	Professional Core (PC)
Approval	37th Meeting of Academic Council, May 2015

LIST OF EXPERIMENTS:

1. Listing applications for mining
2. File format for data mining
3. conversion of various data files
4. Training the given dataset for an application
5. Testing the given dataset for an application
6. Generating accurate models
7. Data pre-processing – data filters
8. Feature selection
9. Web mining
10. Text mining
11. Design of fact & dimension tables
12. Generating graphs for star schema.

DATA WAREHOUSING AND DATA MINING LABORATORY- BCS6L1

LIST OF EXPERIMENTS

	NAME OF THE EXPERIMENT
1	Listing applications for mining
2	File format for data mining
3	Conversion of various data files
4	Training the given dataset for an application
5	Testing the given dataset for an application
6	Generating accurate models
7	Data pre-processing – data filters
8	Feature selection
9	Web mining
10	Text mining
11	Design of fact & dimension tables
12	Generating graphs for star schema.

CONTENT

EX.NO:1

LISTING APPLICATIONS FOR MINING

AIM:

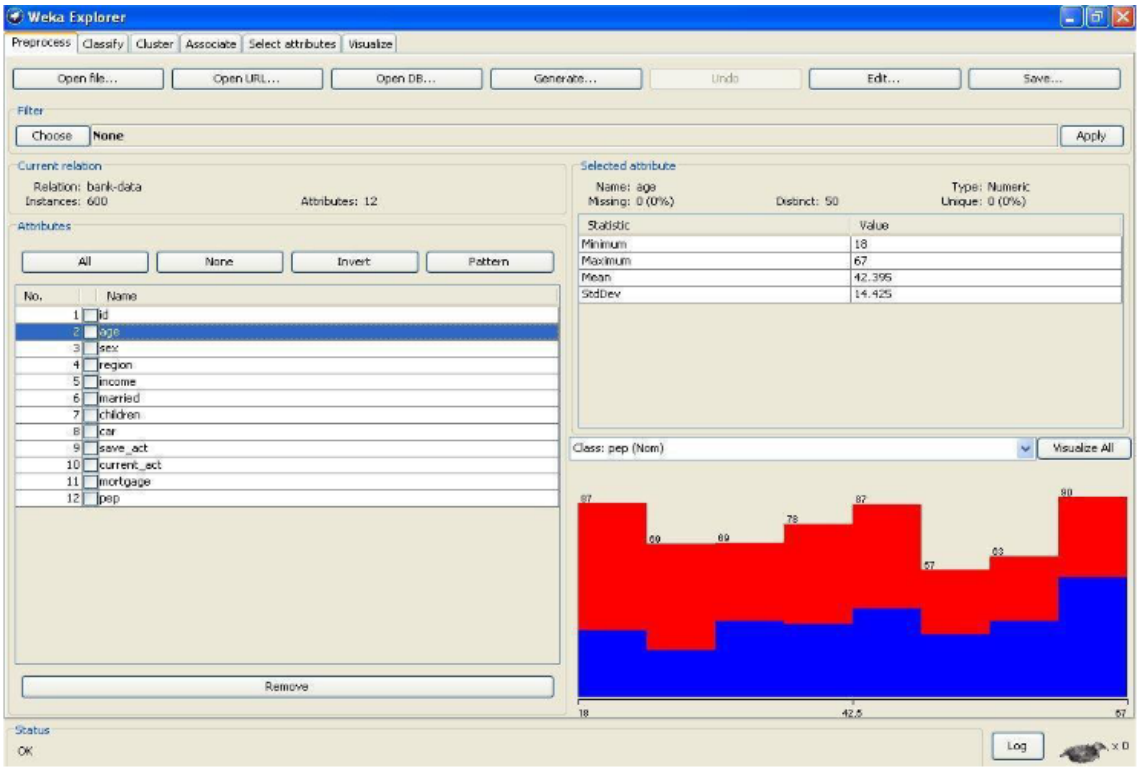
To list all the categorical (or nominal) attributes and the real-valued attributes separately.

RESOURCES: Weka mining tool1.

PROCEDURE:

1)Open the Weka GUI Chooser.
2)Select EXPLORER present in Applications.
3)Select Preprocess Tab.
4)Go to OPEN file and browse the file that is already stored in the system "bank.csv".
5)Clicking on any attribute in the left panel will show the basic statistics on that selected attribute.1.4

OUTPUT:



Result:

Thus the listing applications for the data mining was studied.

EX.NO:2

FILE FORMAT FOR DATA MINING

Aim: To study the file formats for the data mining.

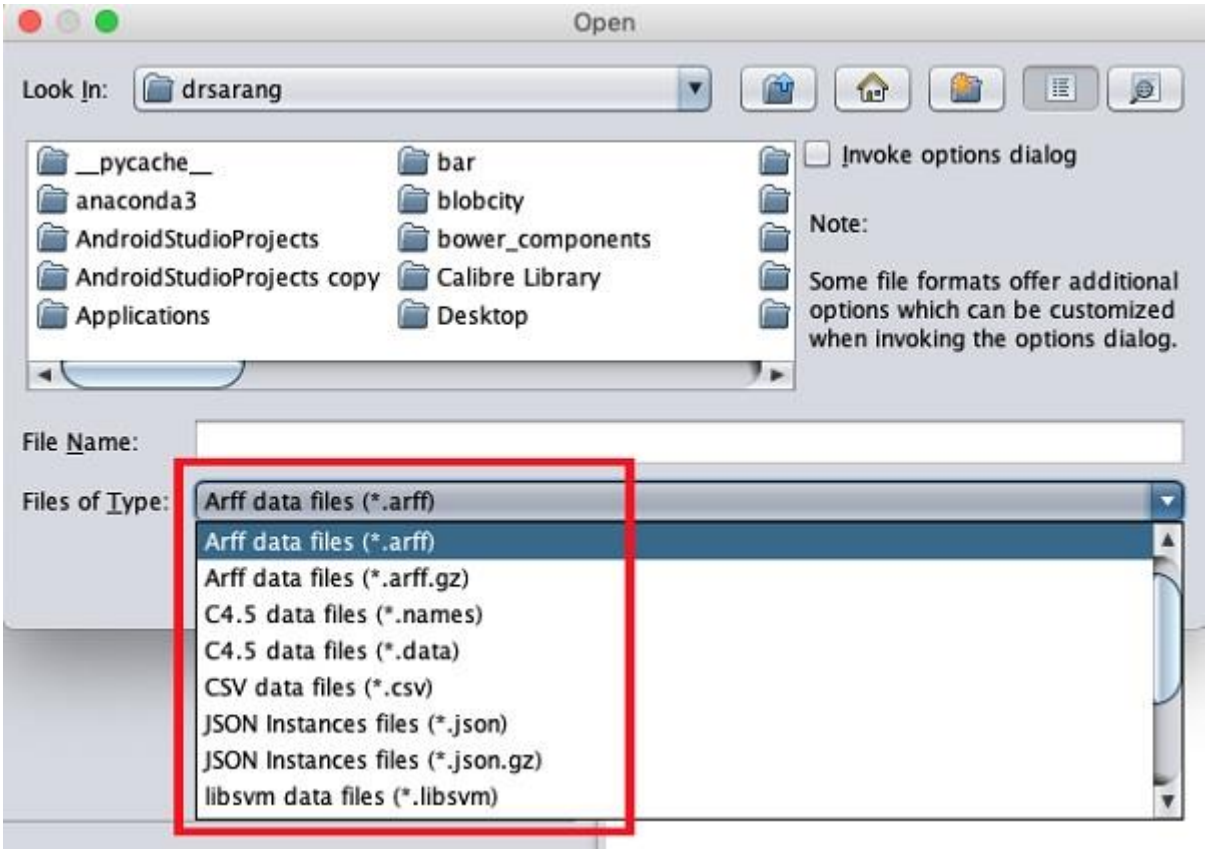
Introduction:

WEKA supports a large number of file formats for the data. The complete list of file formats are given here:

1. arff
2. arff.gz
3. bsi
4. csv
5. dat
6. data
7. json
8. json.gz
9. libsvm
10. m
11. names
12. xrff
13. xrff.gz

The types of files that it supports are listed in the drop-down list box at the bottom of the screen.

This is shown in the screenshot given below.



As you would notice it supports several formats including CSV and JSON.

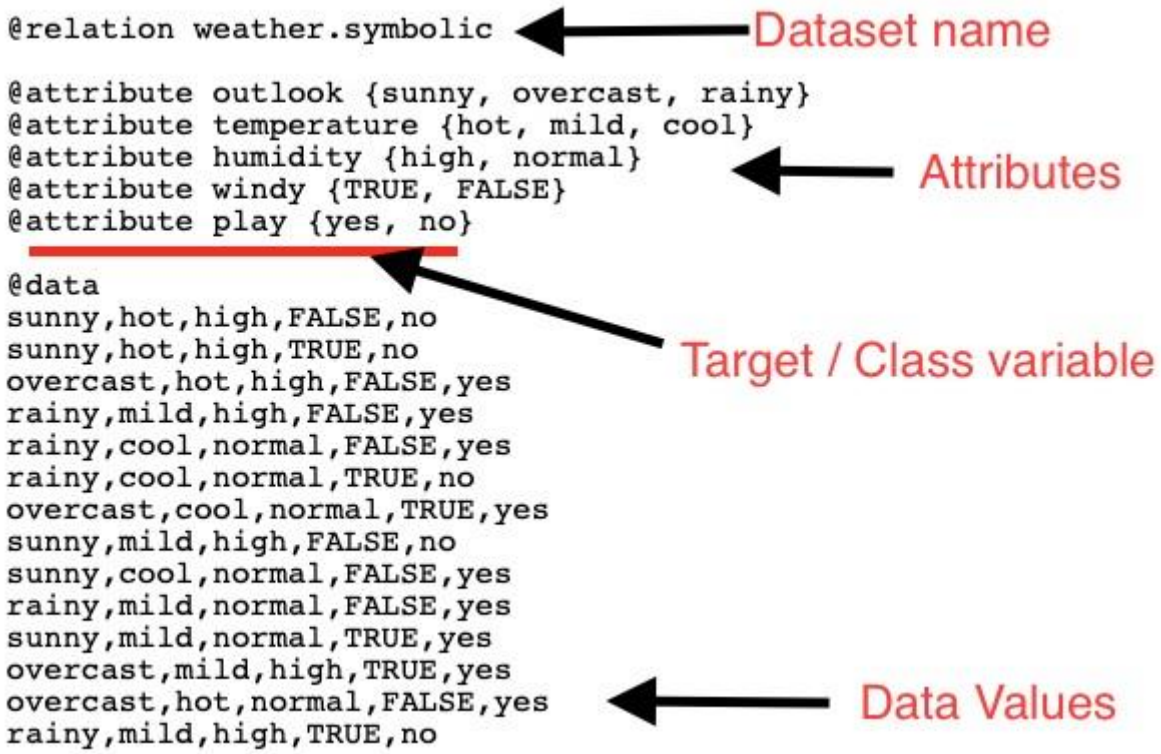
The default file type is Arff.

Arff Format

An Arff file contains two sections - header and data.

The header describes the attribute types.
The data section contains a comma separated list of data.

As an example for Arff format, the Weather data file loaded from the WEKA sample databases is shown below:



From the screenshot, you can infer the following points −

The @relation tag defines the name of the database.

The @attribute tag defines the attributes.

The @data tag starts the list of data rows each containing the comma separated fields.

The attributes can take nominal values as in the case of outlook shown here −

```
@attribute outlook (sunny, overcast, rainy)
```

The attributes can take real values as in this case −

```
@attribute temperature real
```

You can also set a Target or a Class variable called play as shown here −

```
@attribute play (yes, no)
```

The Target assumes two nominal values yes or no.

Result:

Thus the different file formats for the data mining was studied.

EX.NO:3a CONVERSION OF TEXT FILE INTO ARFF FILE

Aim:

To convert a text file to ARFF(Attribute-Relation File Format) using Weka3.8.2 tool.

Objectives:

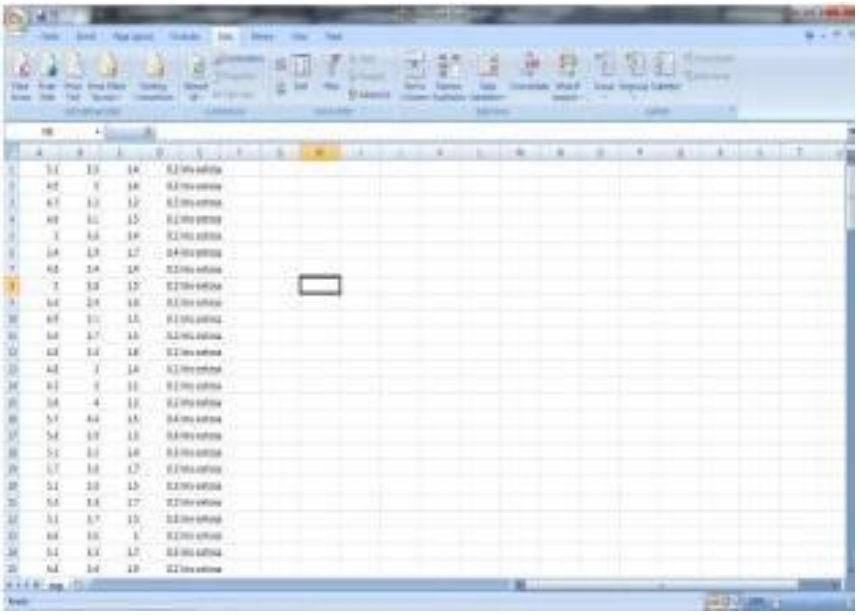
Most of the data that we have collected from public forum is in the text format that cannot be read by Weka tool. Since Weka (Data Mining tool) recognizes the data in ARFF format only we have to convert the text file into ARFF file.

Algorithm:

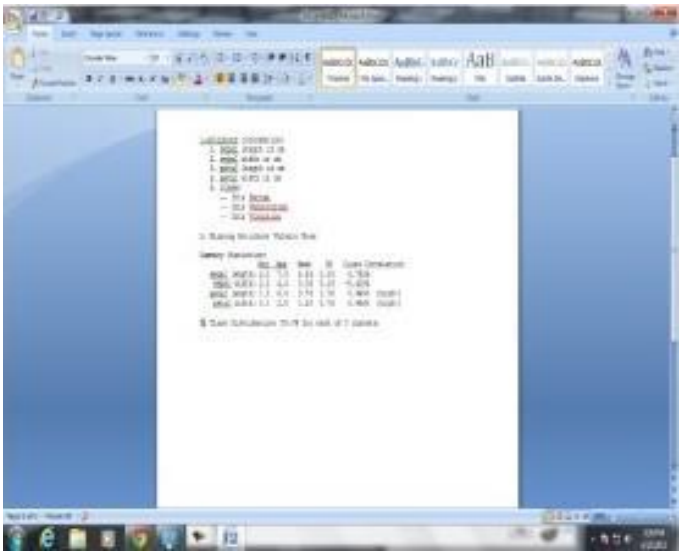
1. Download any data set from UCI data repository.
2. Open the same data file from excel. It will ask for delimiter (which produce column) in excel.
3. Add one row at the top of the data.
4. Enter header for each column.
5. Save file as .CSV (Comma Separated Values) format.
6. Open Weka tool and open the CSV file.
7. Save it as ARFF format.

Output:

Data Text File:



Data ARFF File:



Result:

Thus, conversion of a text file to ARFF(Attribute-Relation File Format) using Weka3.8.2 tool is implemented.

EX.NO:3b. CONVERSION OF ARFF TO TEXT FILE

Aim:

To convert ARFF (Attribute-Relation File Format) into text file.

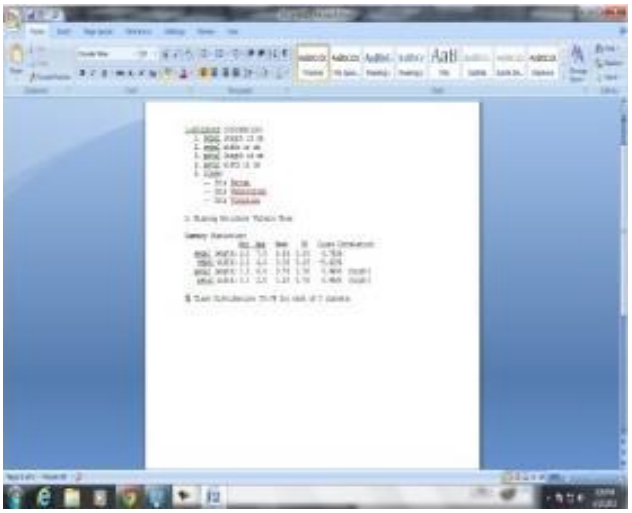
Objectives:

Since the data in the Weka tool is in ARFF file format we have to convert the ARFF file to text format for further processing.

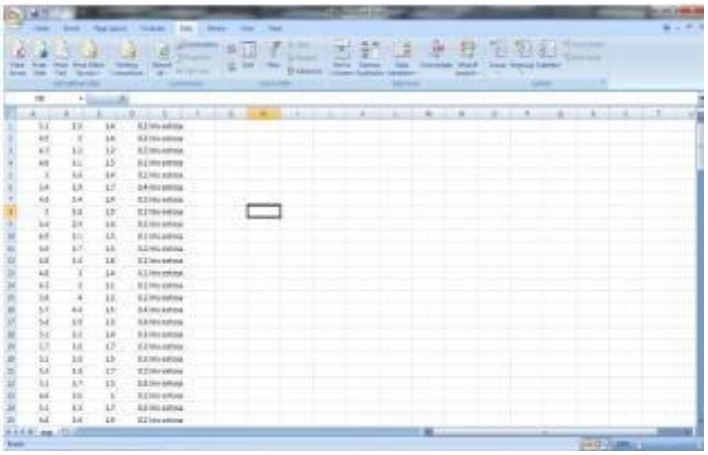
Algorithm:

1. Open any ARFF file in Weka tool.
2. Save the file as CSV format.
3. Open the CSV file in MS-EXCEL.
4. Remove some rows and add coreseponding header to the data.
5. Save it as text file with the desire delimiter.

Data ARFF File:



Data Text File:



Result: Thus conversion of ARFF (Attribute-Relation File Format) into text file is implemented.

EX. No: 4 **TRAINING THE GIVEN DATASET FOR AN APPLICATION**

Aim:

To apply the concept of Linear Regression for training the given dataset.

Algorithm:

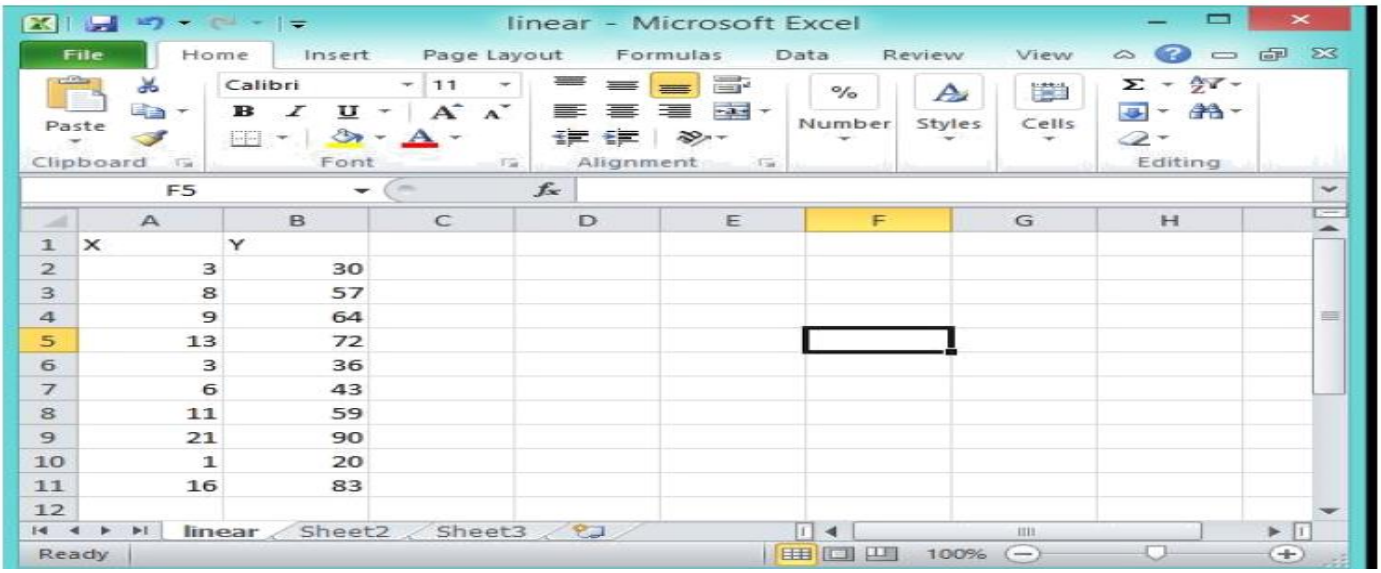
1. Open the weka tool.
2. Download a dataset by using UCI.
3. Apply replace missing values.
4. Apply normalize filter.
5. Click the Classify Tab.
6. Choose the Simple Linear Regression option.
7. Select the training set of data.
8. Start the validation process.
9. Note the output.

LINEAR REGRESSION:

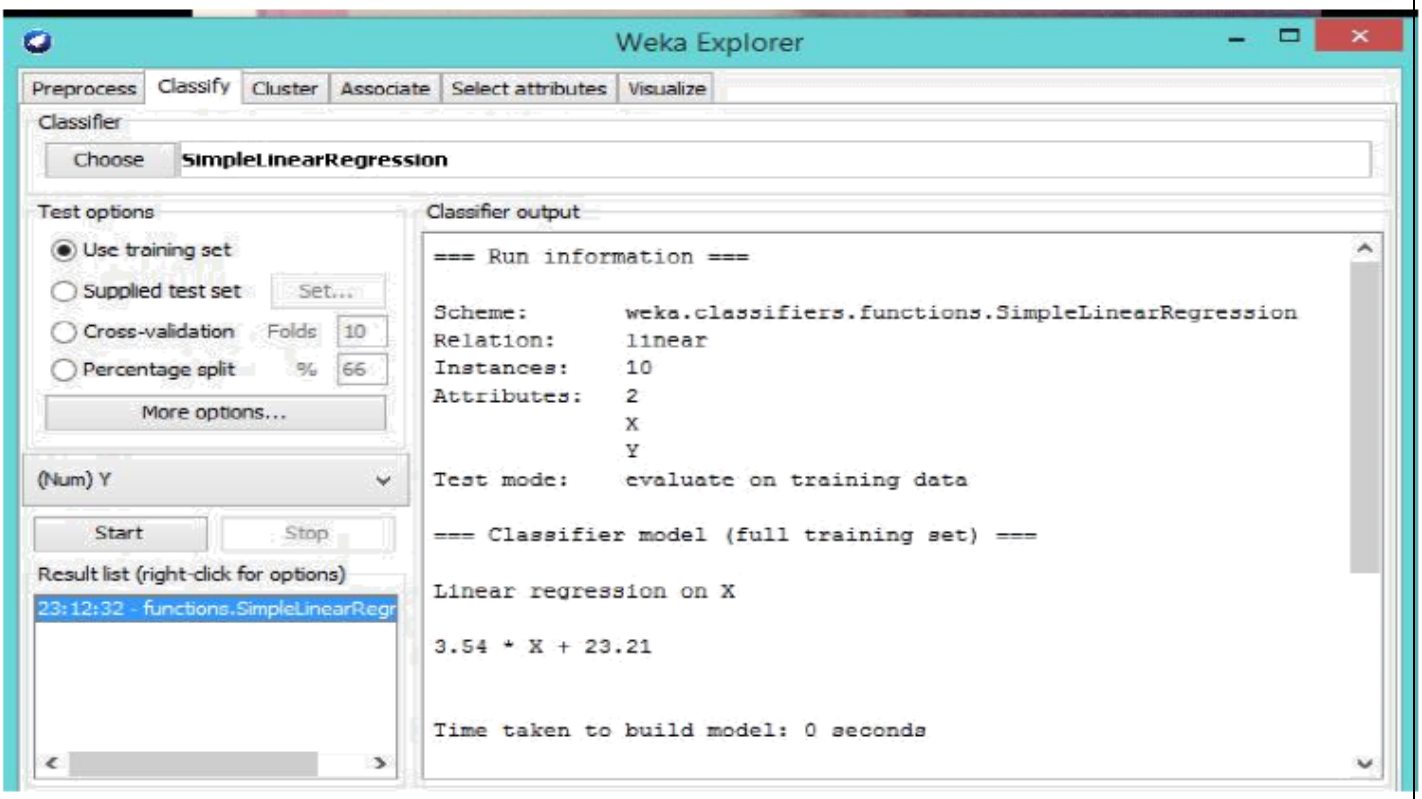
In statistics, Linear Regression is an approach for modeling a relationship between a scalar dependent variable Y and one or more explanatory variables denoted X.the case of explanatory variable is called Simple Linear Regression.
Coefficient of Linear Regression is given by: **Y=ax+b**

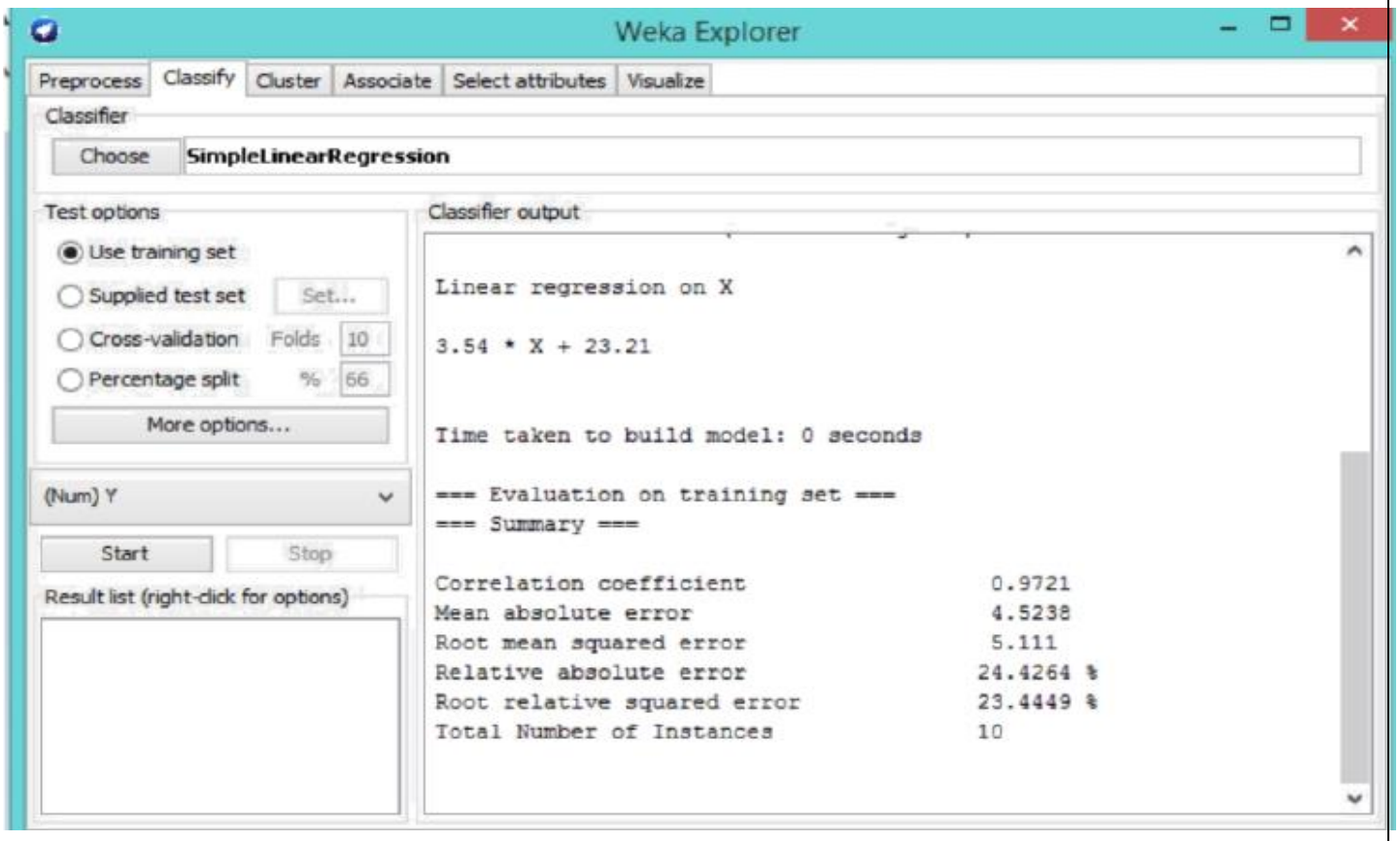
PROBLEM:
Consider the dataset below where x is the number of working expeince of a college graduate and y is the corresponding salary of the graduate. Build a regression equation and predict the salary of college graduate whose experience is 10 years.

Input:



Output:





Result: Thus the concept of Linear Regression for training the given dataset is applied and implemented.

EX. No: 5 TESTING THE GIVEN DATASET FOR AN APPLICATION

Aim:

To apply the Navie Bayes Classification for testing the given dataset.

Algorithm:

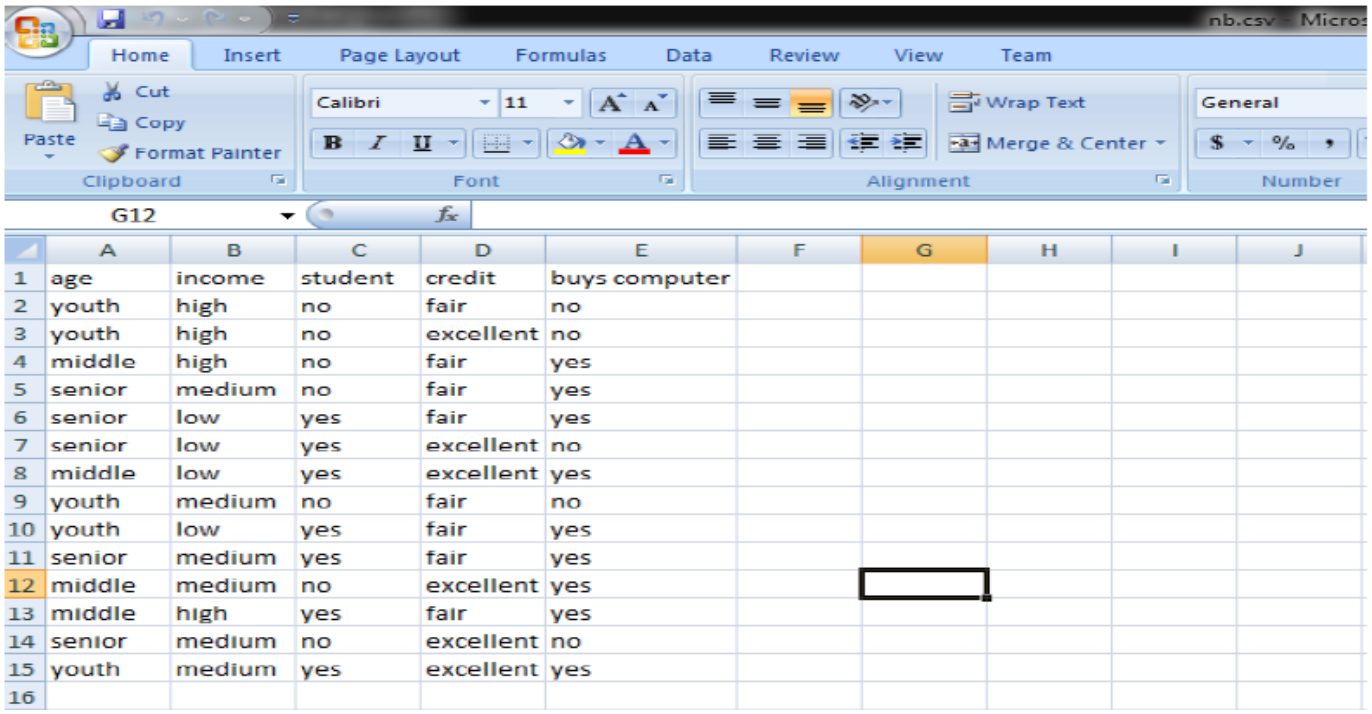
1. Open the weka tool.
2. Download a dataset by using UCI.
3. Apply replace missing values.
4. Apply normalize filter.
5. Click the Classification Tab.
6. Apply Navie Bayes Classification.
7. Find the Classified Value.
8. Note the output.

Bayes' Theorem In the Classification Context:

X is a data tuple. In Bayesian term it is considered "evidence".H is some hypothesis that X belongs to a specified class C .P(H|X) is the posterior probability of H conditioned on X .

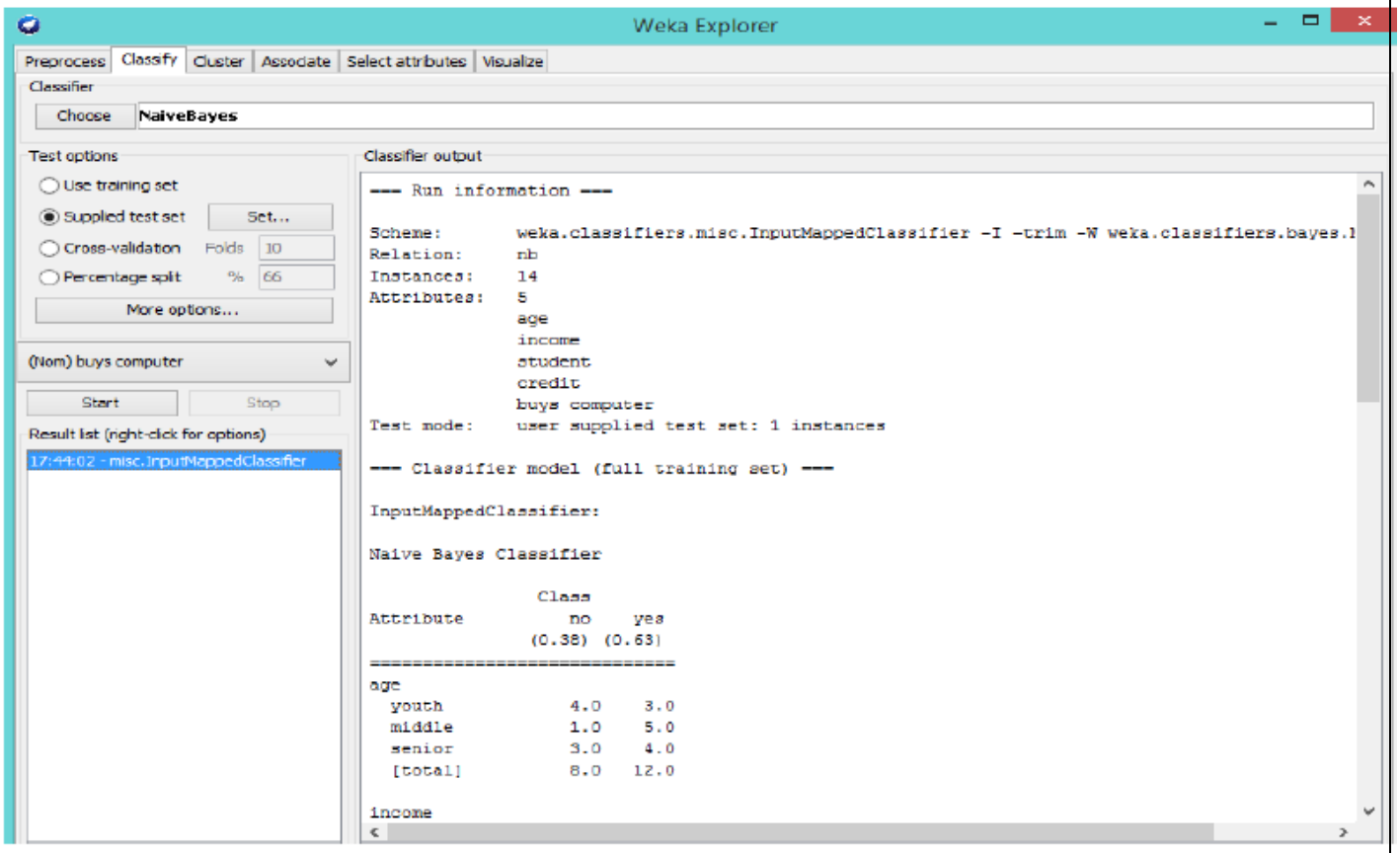
Example: predict whether a costumer will buy a computer or not " Costumers are described by two attributes: age and income " X is a 35 years-old costumer with an income of 40k " H is the hypothesis that the costumer will buy a computer " P(H|X) reflects the probability that costumer X will buy a computer given that we know the costumers' age and income.

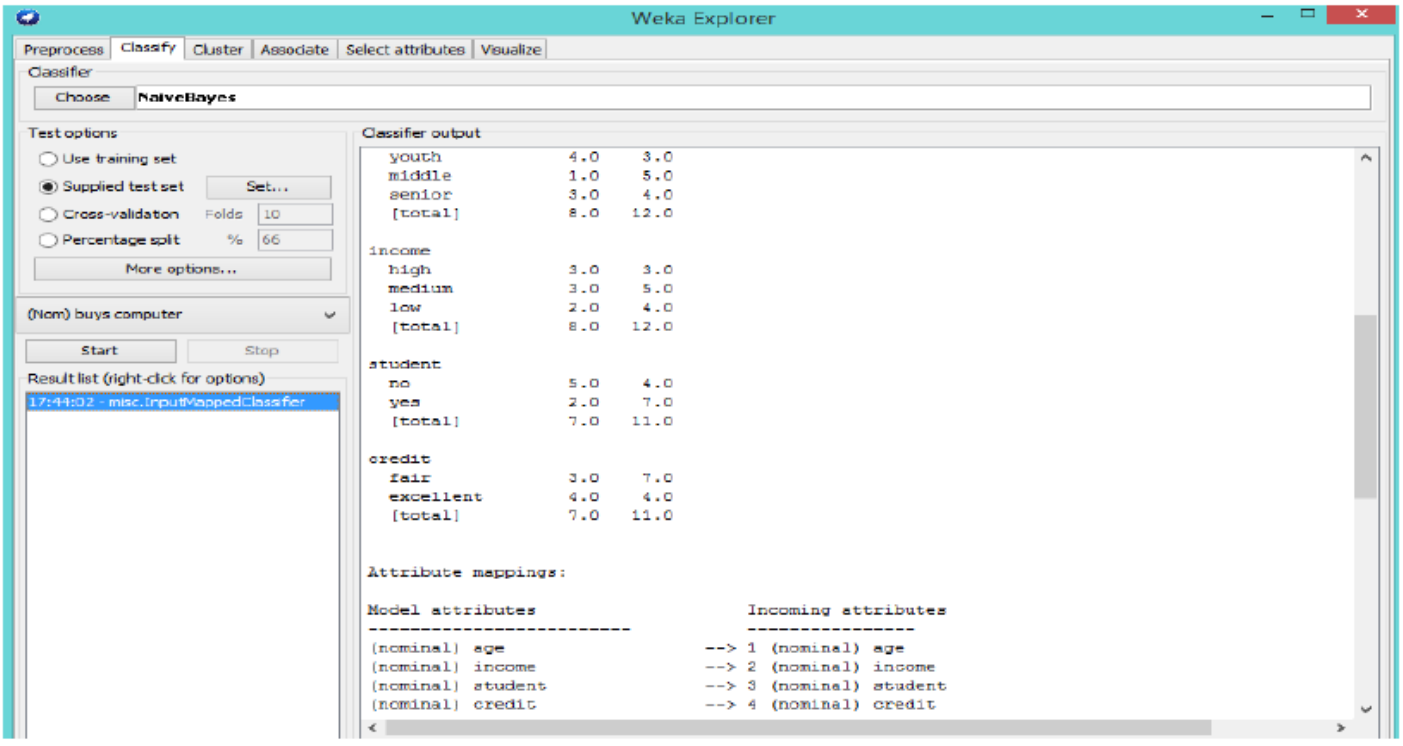
Input Data:

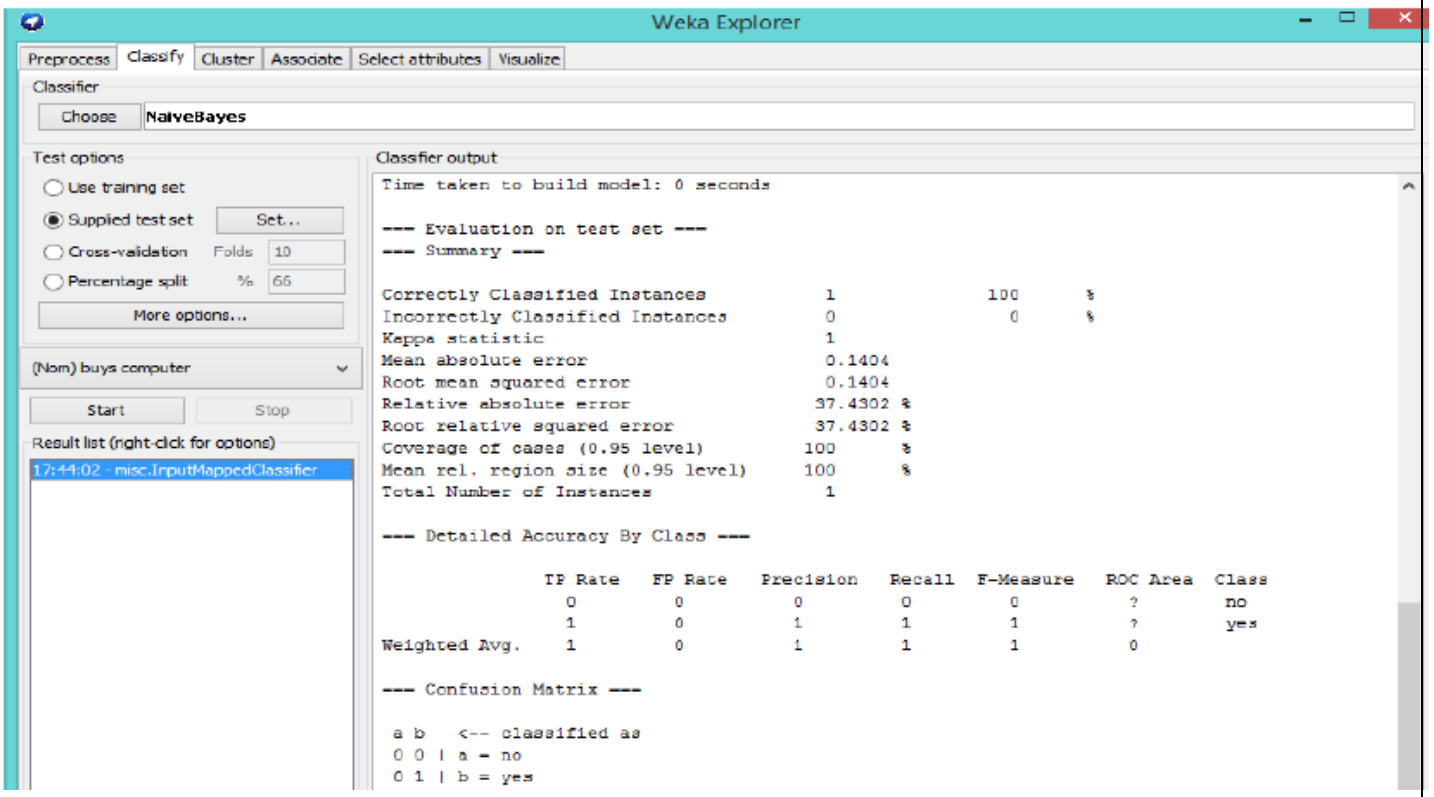


	A	B	C	D	E	F	G	H	I	J
1	age	income	student	credit	buys computer					
2	youth	high	no	fair	no					
3	youth	high	no	excellent	no					
4	middle	high	no	fair	yes					
5	senior	medium	no	fair	yes					
6	senior	low	yes	fair	yes					
7	senior	low	yes	excellent	no					
8	middle	low	yes	excellent	yes					
9	youth	medium	no	fair	no					
10	youth	low	yes	fair	yes					
11	senior	medium	yes	fair	yes					
12	middle	medium	no	excellent	yes					
13	middle	high	yes	fair	yes					
14	senior	medium	no	excellent	no					
15	youth	medium	yes	excellent	yes					
16										

Output data:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
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)
17:44:02 - misc.InputMappedClassifier
Classifier output
=== Run information ===
Scheme: weka.classifiers.misc.InputMappedClassifier -I -trim -N weka.classifiers.bayes.
Relation: nb
Instances: 14
Attributes: 5
age
income
student
credit
buys computer
Test mode: user supplied test set: 1 instances
=== Classifier model (full training set) ===
InputMappedClassifier:
Naive Bayes Classifier
Class
Attribute no yes
(0.38) (0.63)
age
youth 4.0 3.0
middle 1.0 5.0
senior 3.0 4.0
[total] 8.0 12.0
income


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
NaiveBayes
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)
17:44:02 - misc.InputMappedClassifier
Classifier output
youth 4.0 3.0
middle 1.0 5.0
senior 3.0 4.0
[total] 8.0 12.0
income
high 3.0 3.0
medium 3.0 5.0
low 2.0 4.0
[total] 8.0 12.0
student
no 5.0 4.0
yes 2.0 7.0
[total] 7.0 11.0
credit
fair 3.0 7.0
excellent 4.0 4.0
[total] 7.0 11.0
Attribute mappings:
Model attributes Incoming attributes
(nominal) age --> 1 (nominal) age
(nominal) income --> 2 (nominal) income
(nominal) student --> 3 (nominal) student
(nominal) credit --> 4 (nominal) credit



Result:

Thus the Navie Bayes Classification for testing the given dataset is implemented.

EX. No: 6 GENERATE ACCURATE MODEL

Aim:

To find the good result (by improving the performance) using the training set and testing data set for numerical values.

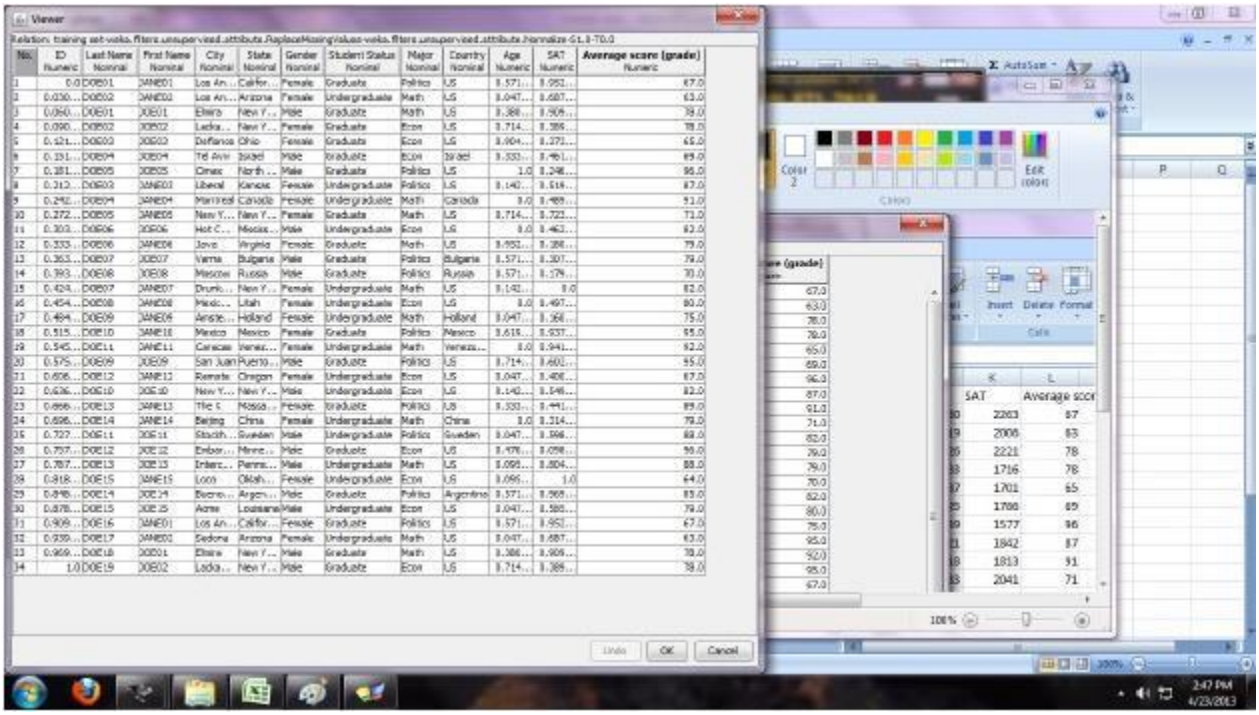
Objectives:

To develop training and testing data using numerical data set in order to get accurate model for classification.

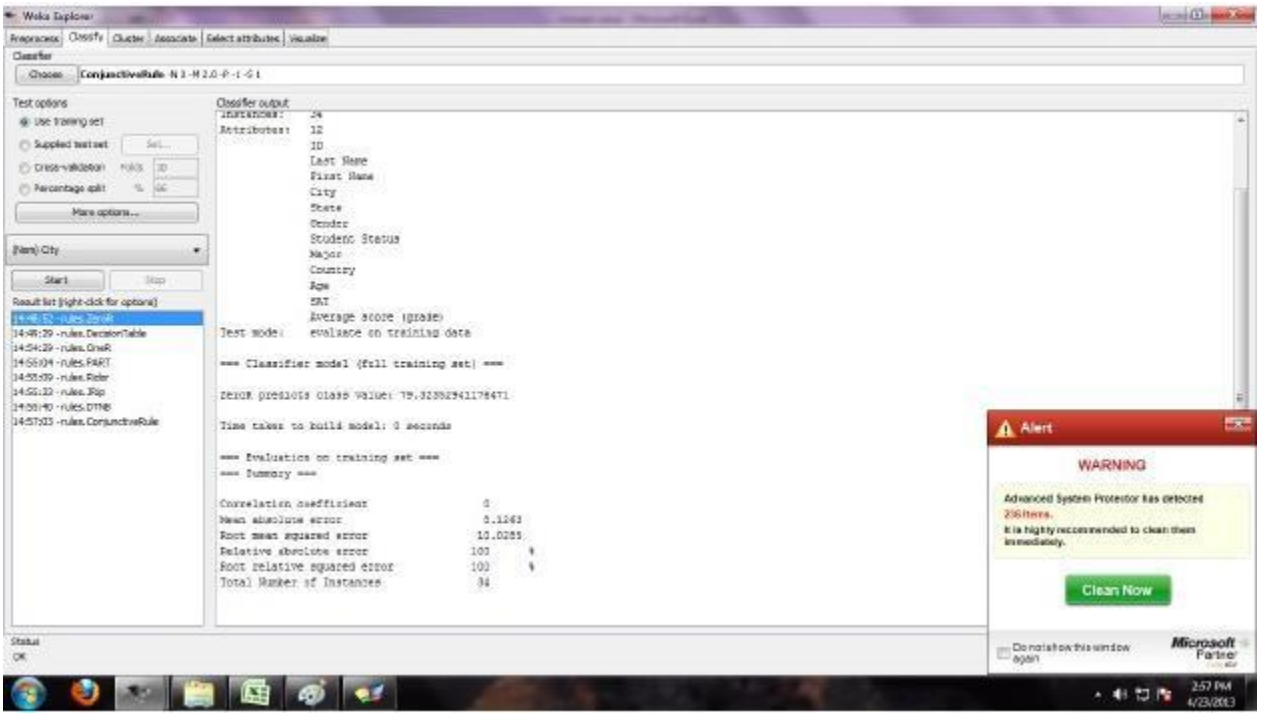
A**LGORITHM:**

1. Download any data set.
2. Save the file with .ARFF format.
3. Apply ‘Replace Missing Values’ filter.
4. Normalize the values by applying normalize flter.
5. Go to unsupervised instance remove percentage
6. Right click on that (show properties) option then select 70% true and save it as training.arff
7. Select the original data set then right click on show properties then give 70% false and save it as testing.arff
8. Select classification and apply various algorithms.

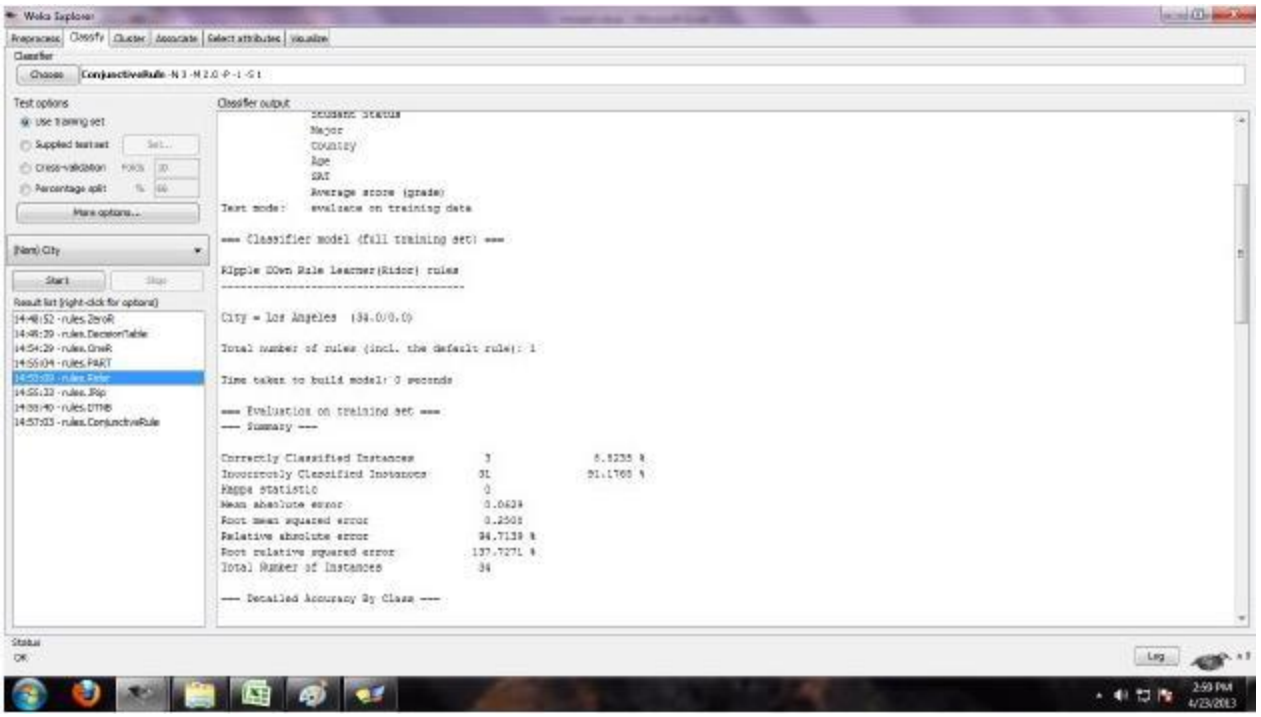
TRAINING DATA:



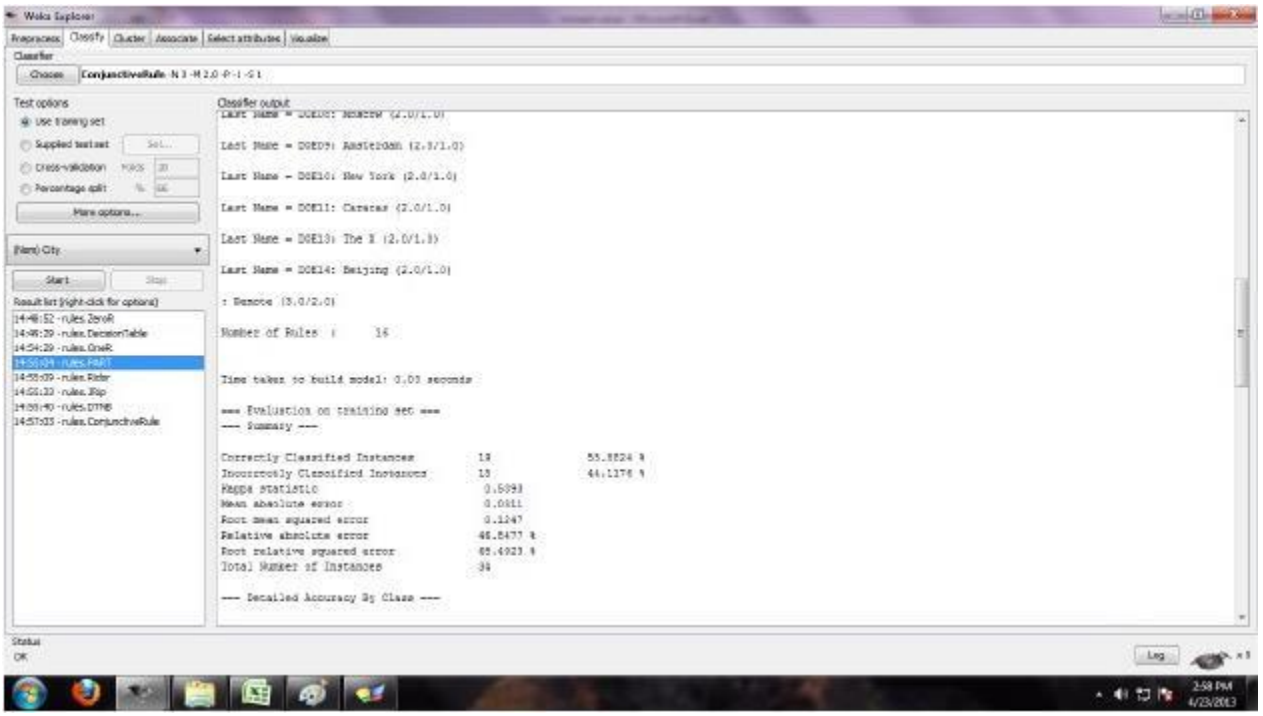
ZeroR:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:39 - rules.Ridor
14:55:22 - rules.JRip
14:59:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
Instances: 34
Attributes: 12
ID
Last Name
First Name
City
State
Gender
Student Status
Major
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
ZeroR predicts class value: 79.32352941176471
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correlation coefficient 0
Mean absolute error 6.1245
Root mean squared error 10.0285
Relative absolute error 100 %
Root relative squared error 100 %
Total Number of Instances 34
Alert
WARNING
Advanced System Protector has detected
256 Items.
It is highly recommended to clean them immediately.
Clean Now
Do not show this window again
Microsoft Partner
Status
OK
2:57 PM
4/23/2013

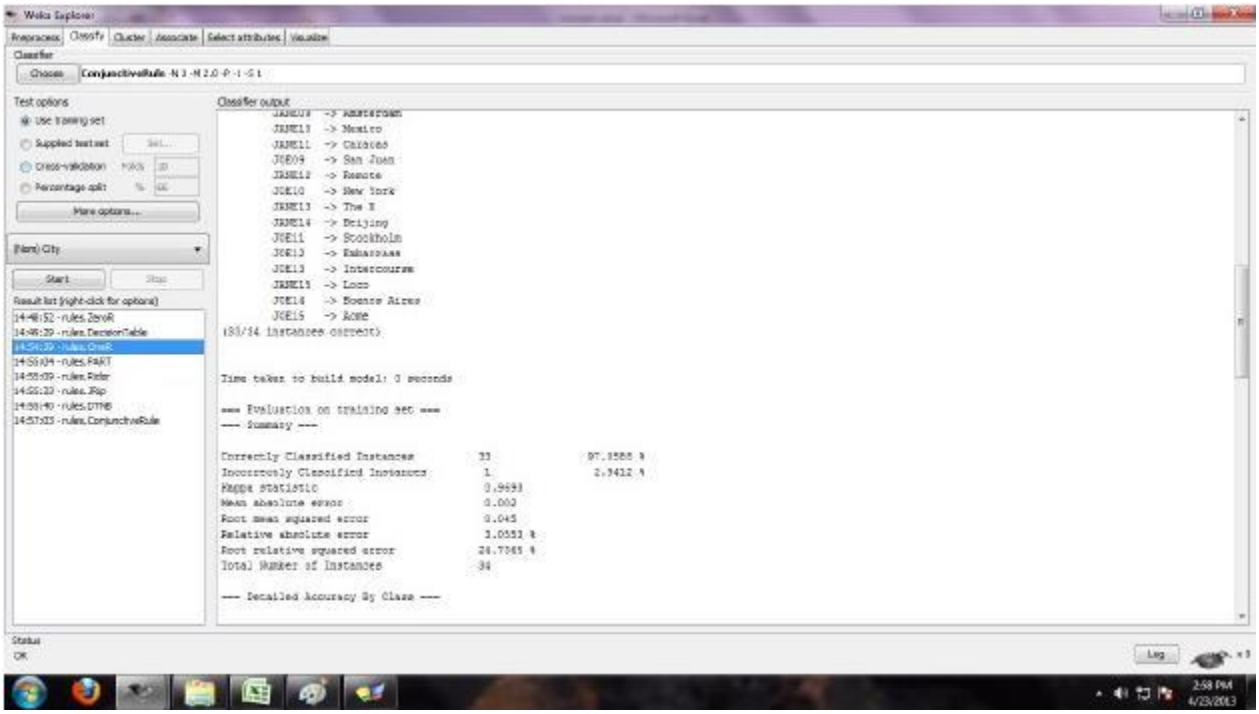
Ridor:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:39 - rules.Ridor
14:55:22 - rules.JRip
14:59:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
Student Status
Major
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
Ripple DOwn Rule Learner(Ridor) rules
City = Los Angeles (34.0/0.0)
Total number of rules (incl. the default rule): 1
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 3 8.8235 %
Incorrectly Classified Instances 31 91.1765 %
Kappa statistic 0
Mean absolute error 0.0629
Root mean squared error 0.2508
Relative absolute error 94.7139 %
Root relative squared error 137.7271 %
Total Number of Instances 34
=== Detailed Accuracy By Class ===
Status
OK
Log
2:59 PM
4/23/2013

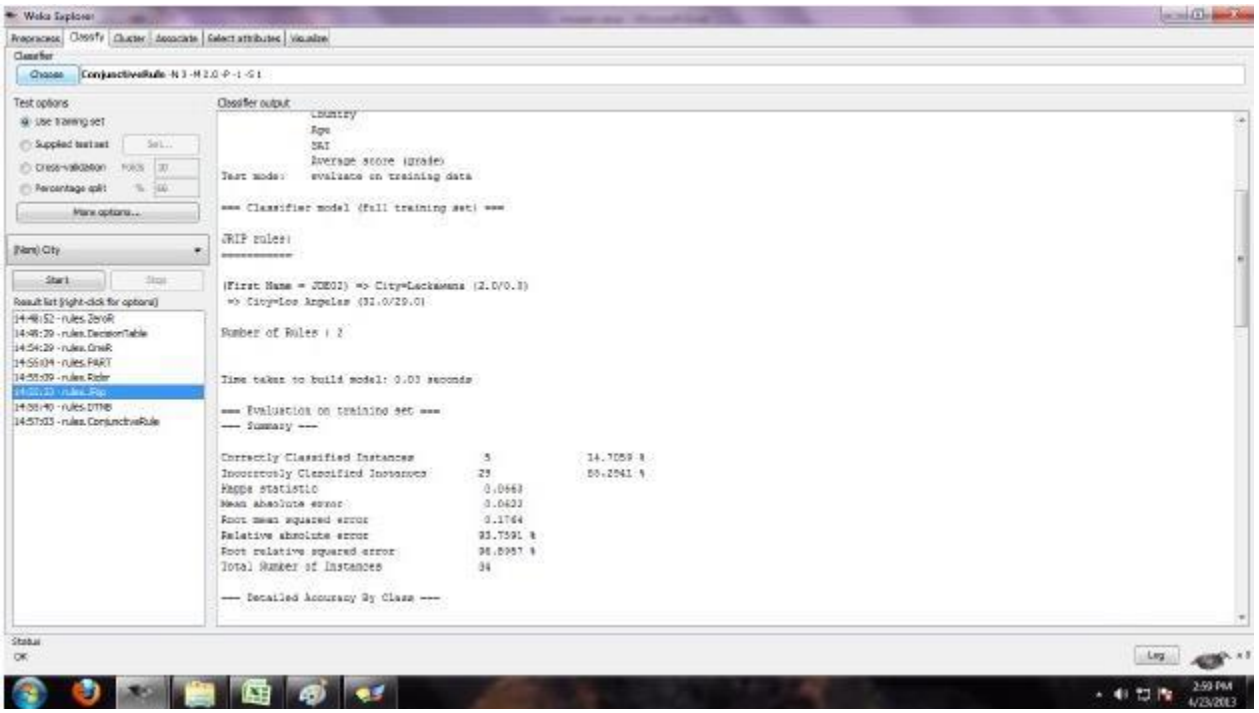
PART:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:49:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:56:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
Last Name = DOE9: Amsterdam (2.3/1.0)
Last Name = DOE10: New York (2.0/1.0)
Last Name = DOE11: Caracas (2.0/1.0)
Last Name = DOE13: The X (2.0/1.3)
Last Name = DOE14: Beijing (2.0/1.0)
: Remove (3.0/2.0)
Number of Rules : 16
Time taken to build model: 0.03 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 18 55.8824 %
Incorrectly Classified Instances 13 44.1176 %
Kappa statistic 0.5093
Mean absolute error 0.0811
Root mean squared error 0.1247
Relative absolute error 46.8477 %
Root relative squared error 65.4923 %
Total Number of Instances 34
=== Detailed Accuracy By Class ===
Status
OK
Log
x 1
2:58 PM
4/23/2013

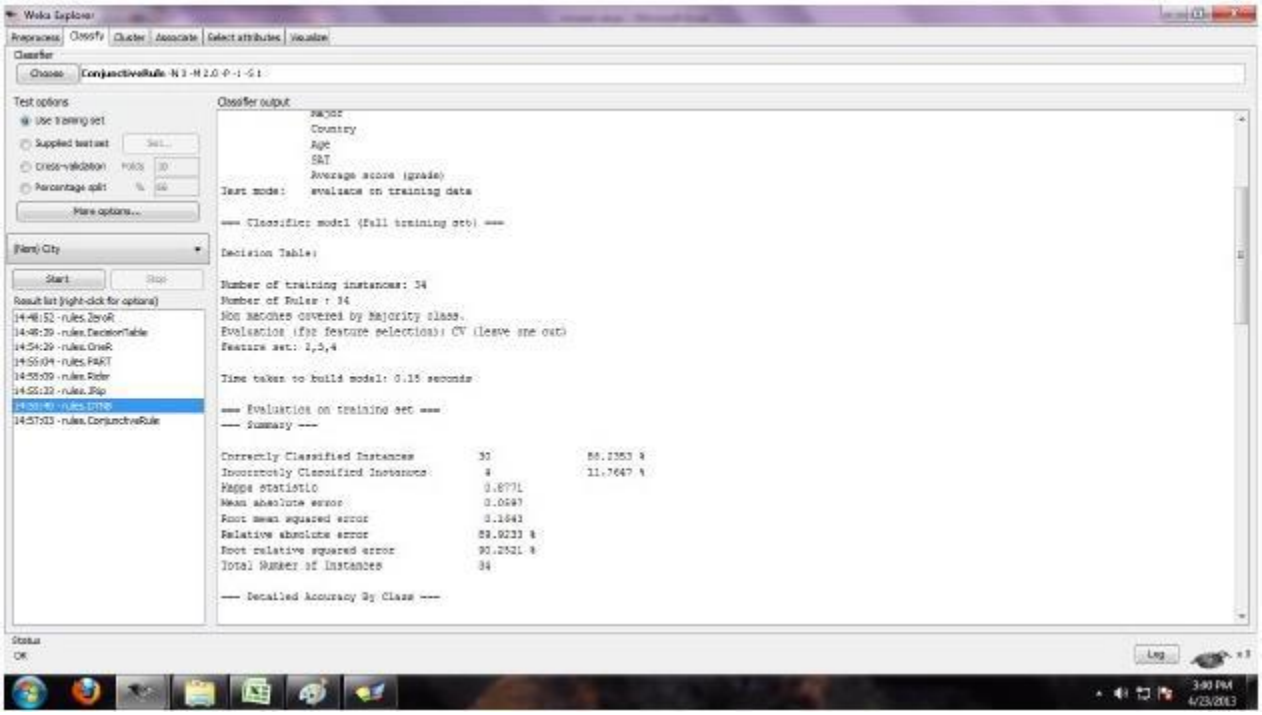
OneR:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:49:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:56:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
JANE13 -> Mexico
JANE11 -> Caracas
JOE09 -> San Juan
JANE12 -> Remote
JOE10 -> New York
JANE13 -> The X
JANE14 -> Beijing
JOE11 -> Stockholm
JOE13 -> Embarcaas
JOE13 -> Intercourse
JANE13 -> Loco
JOE14 -> Buenos Aires
JOE15 -> Acme
(33/34 instances correct)
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 33 97.0588 %
Incorrectly Classified Instances 1 2.9412 %
Kappa statistic 0.9693
Mean absolute error 0.002
Root mean squared error 0.045
Relative absolute error 1.0553 %
Root relative squared error 24.7365 %
Total Number of Instances 34
=== Detailed Accuracy By Class ===
Status
OK
Log
x 1
2:58 PM
4/23/2013

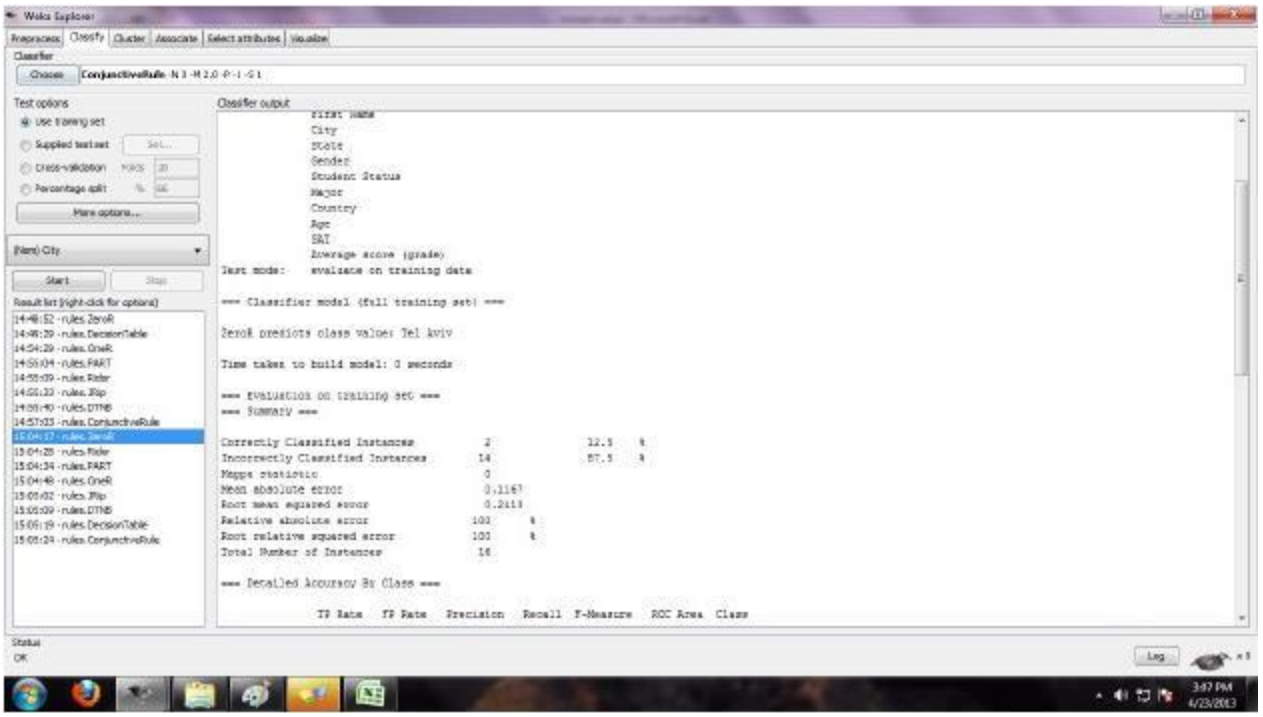
JRip:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:39 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:56:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
JRIP rules:
(First Name = JOE02) => City=Lackawana (2.0/0.3)
=> City=Los Angeles (32.0/29.0)
Number of Rules : 2
Time taken to build model: 0.03 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 5 14.7059 %
Incorrectly Classified Instances 29 85.2941 %
Kappa statistic 0.0663
Mean absolute error 0.0422
Root mean squared error 0.1764
Relative absolute error 93.7591 %
Root relative squared error 96.8957 %
Total Number of Instances 34
=== Detailed Accuracy By Class ===
Status
OK
Log
x 0
2:59 PM
4/23/2013

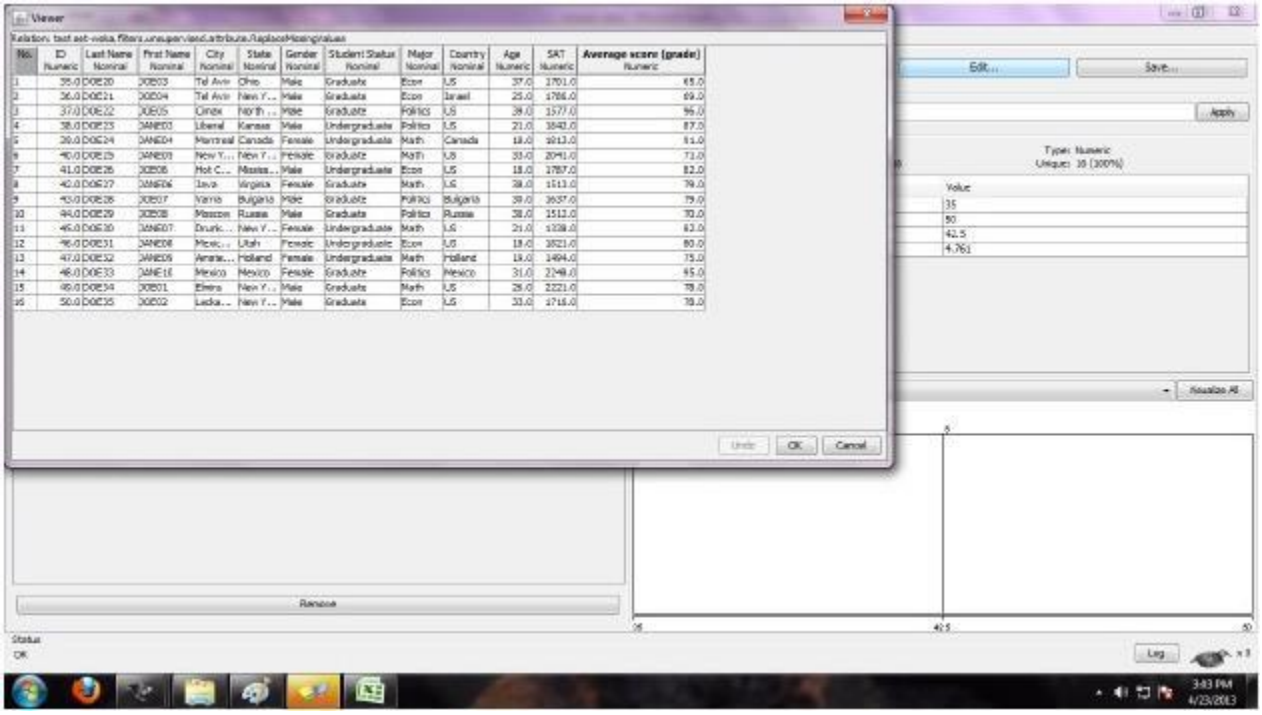
DTNB:


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation
Percentage split
More options...
(Nom) City
Start
Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:39 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:56:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
Classifier output
Major
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
Decision Table:
Number of training instances: 34
Number of Rules : 34
Non matches covered by Majority class.
Evaluation (for feature selection): CV (leave one out)
Feature set: 1,3,4
Time taken to build model: 0.15 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 30 88.2353 %
Incorrectly Classified Instances 4 11.7647 %
Kappa statistic 0.8771
Mean absolute error 0.0697
Root mean squared error 0.1843
Relative absolute error 69.0233 %
Root relative squared error 90.2521 %
Total Number of Instances 34
=== Detailed Accuracy By Class ===
Status
OK
Log
x 0
3:00 PM
4/23/2013

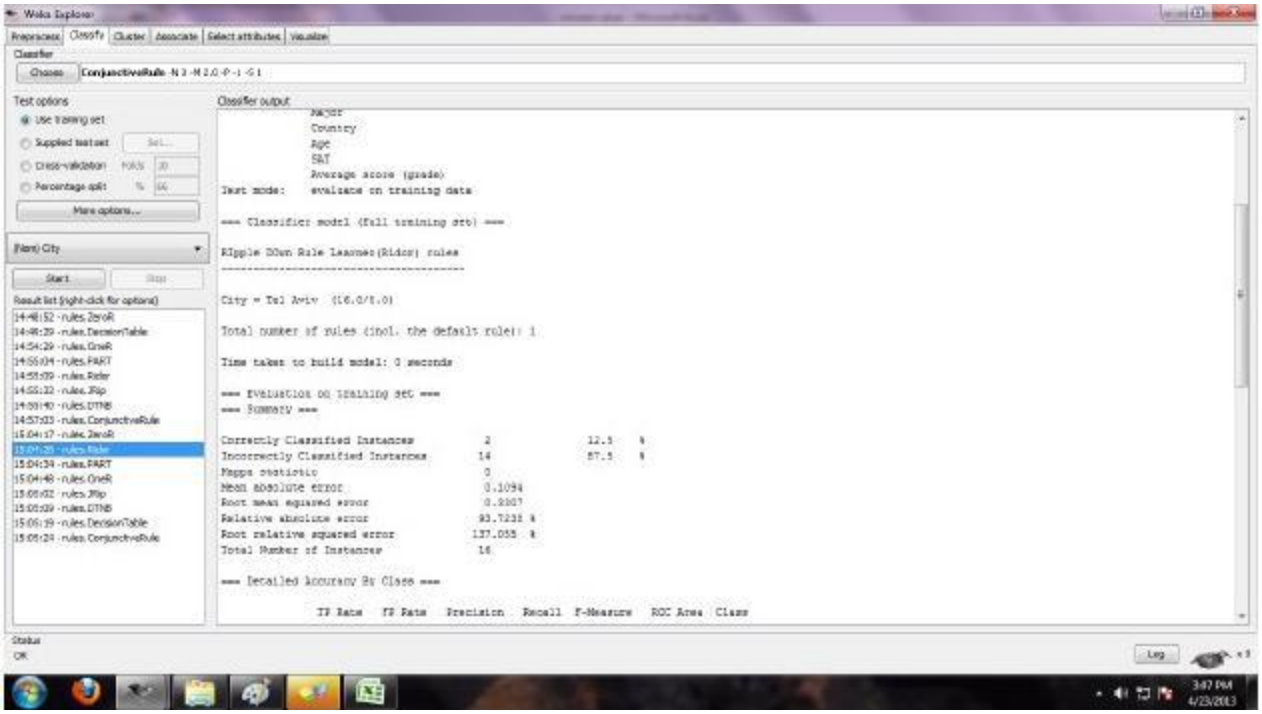
TEST DATA:



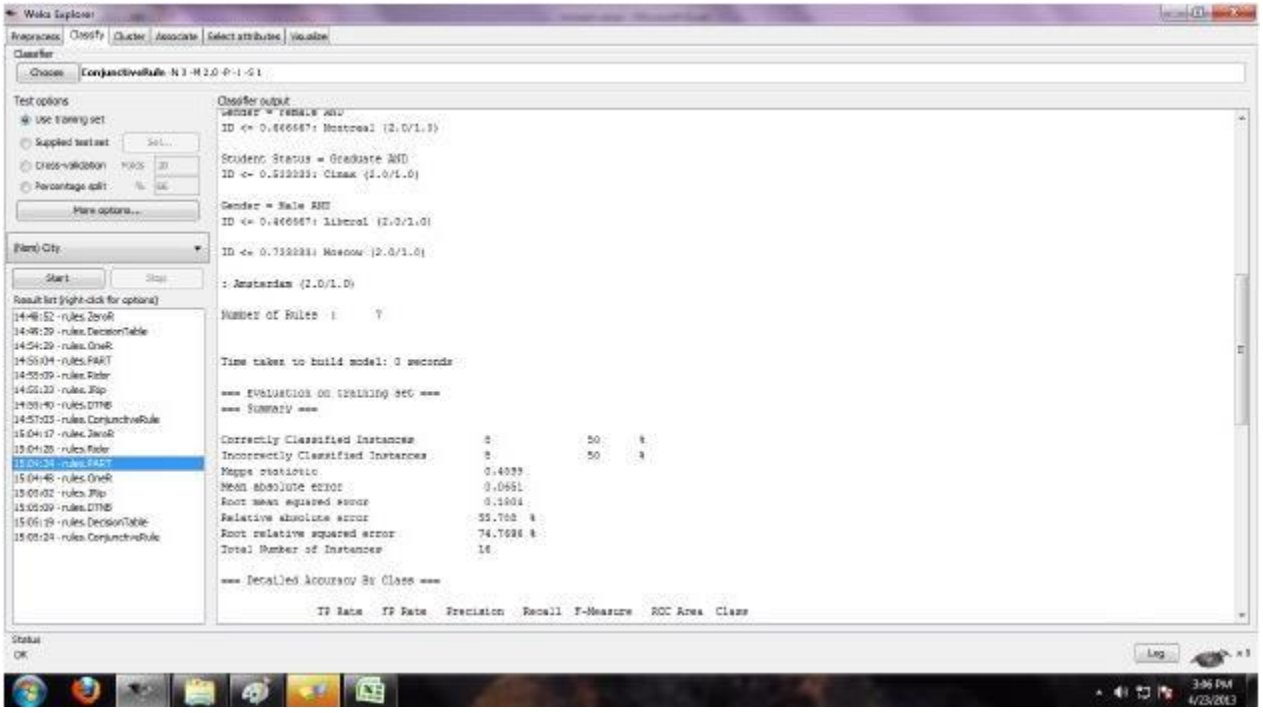
ZeroR:



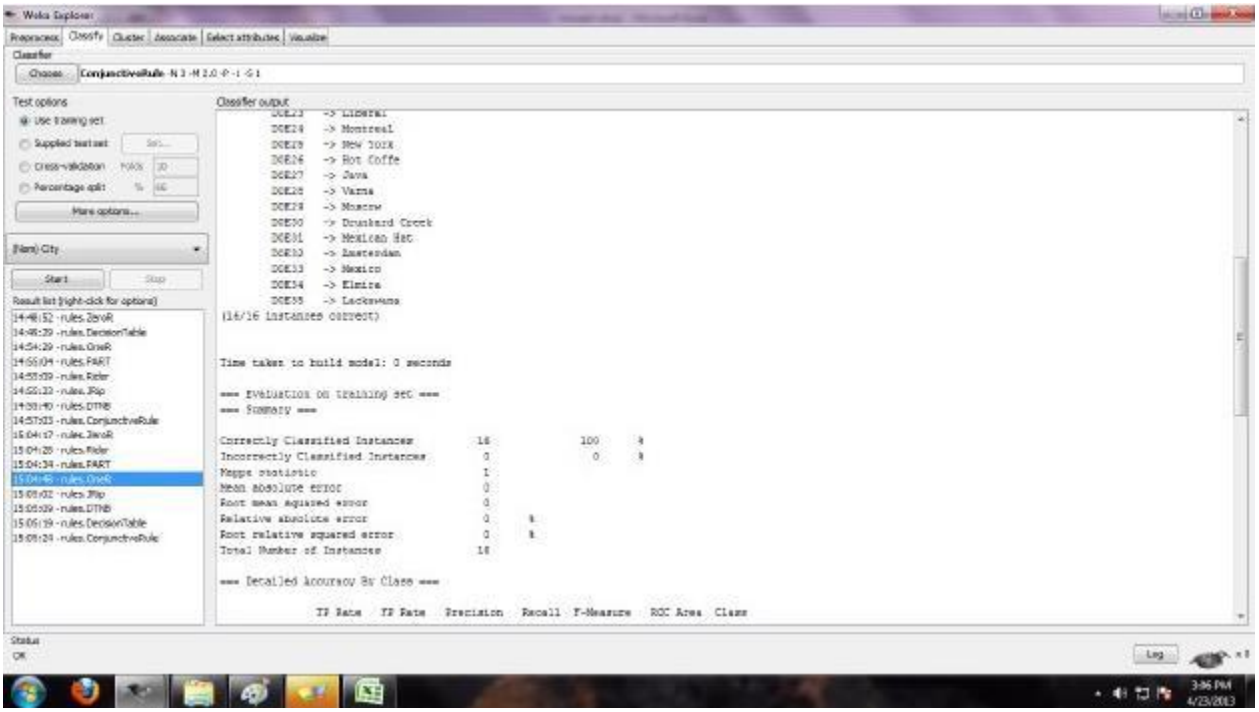
Ridor:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) City
Start Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:59:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
15:04:17 - rules.ZeroR
15:04:25 - rules.Ridor
15:04:34 - rules.PART
15:04:48 - rules.OneR
15:05:02 - rules.JRip
15:05:09 - rules.DTNB
15:05:19 - rules.DecisionTable
15:05:24 - rules.ConjunctiveRule
Classifier output
Major
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
RIpple DOwn Rule Learner(Ridor) rules
City = Tel Aviv (16.0/8.0)
Total number of rules (incl. the default rule): 1
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 2 12.5 %
Incorrectly Classified Instances 14 87.5 %
Kappa statistic 0
Mean absolute error 0.1094
Root mean squared error 0.2807
Relative absolute error 93.7238 %
Root relative squared error 137.055 %
Total Number of Instances 16
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
Status
OK
Log
x 1
3:47 PM
4/23/2013

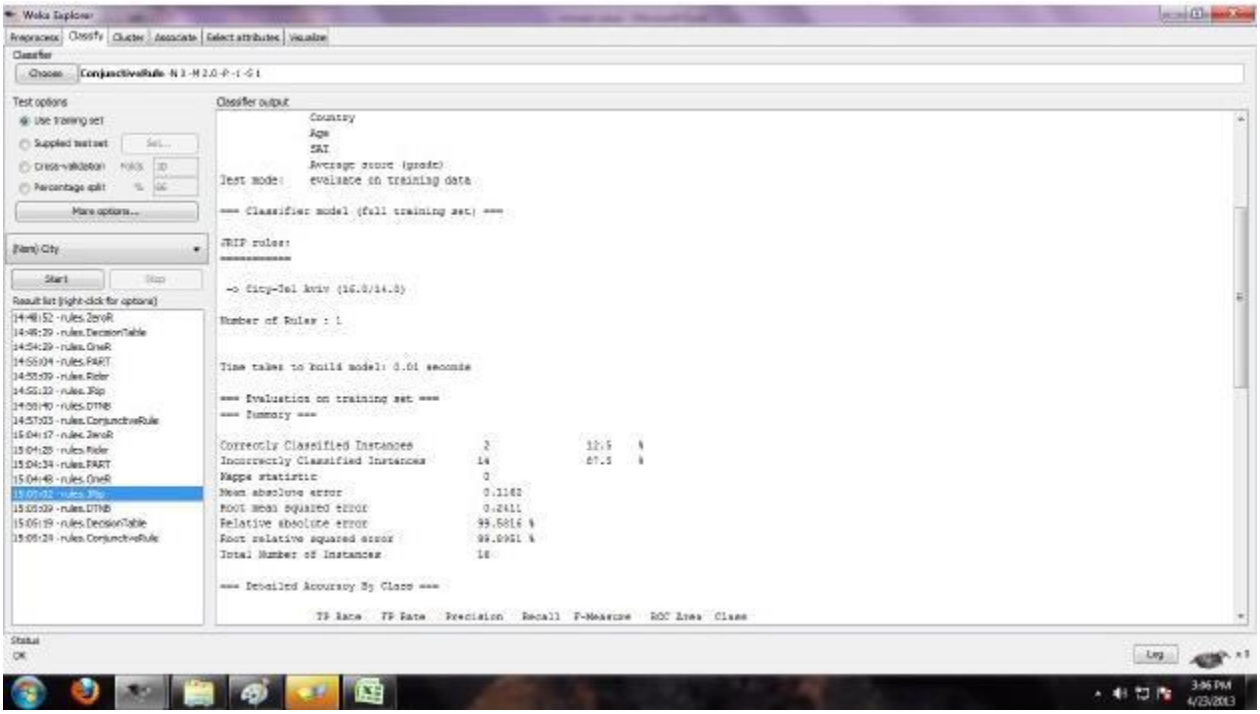
PART:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) City
Start Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:48:29 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:09 - rules.Ridor
14:55:33 - rules.JRip
14:59:40 - rules.DTNB
14:57:03 - rules.ConjunctiveRule
15:04:17 - rules.ZeroR
15:04:25 - rules.Ridor
15:04:34 - rules.PART
15:04:48 - rules.OneR
15:05:02 - rules.JRip
15:05:09 - rules.DTNB
15:05:19 - rules.DecisionTable
15:05:24 - rules.ConjunctiveRule
Classifier output
ID <= 0.666667: Montreal (2.0/1.0)
Student Status = Graduate AND
ID <= 0.533333: Cimax (2.0/1.0)
Gender = Male AND
ID <= 0.466667: Liberal (2.0/1.0)
ID <= 0.733333: Moscow (2.0/1.0)
: Amsterdam (2.0/1.0)
Number of Rules : 7
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 8 50 %
Incorrectly Classified Instances 8 50 %
Kappa statistic 0.4889
Mean absolute error 0.0651
Root mean squared error 0.1804
Relative absolute error 55.768 %
Root relative squared error 74.7698 %
Total Number of Instances 16
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
Status
OK
Log
x 1
3:46 PM
4/23/2013

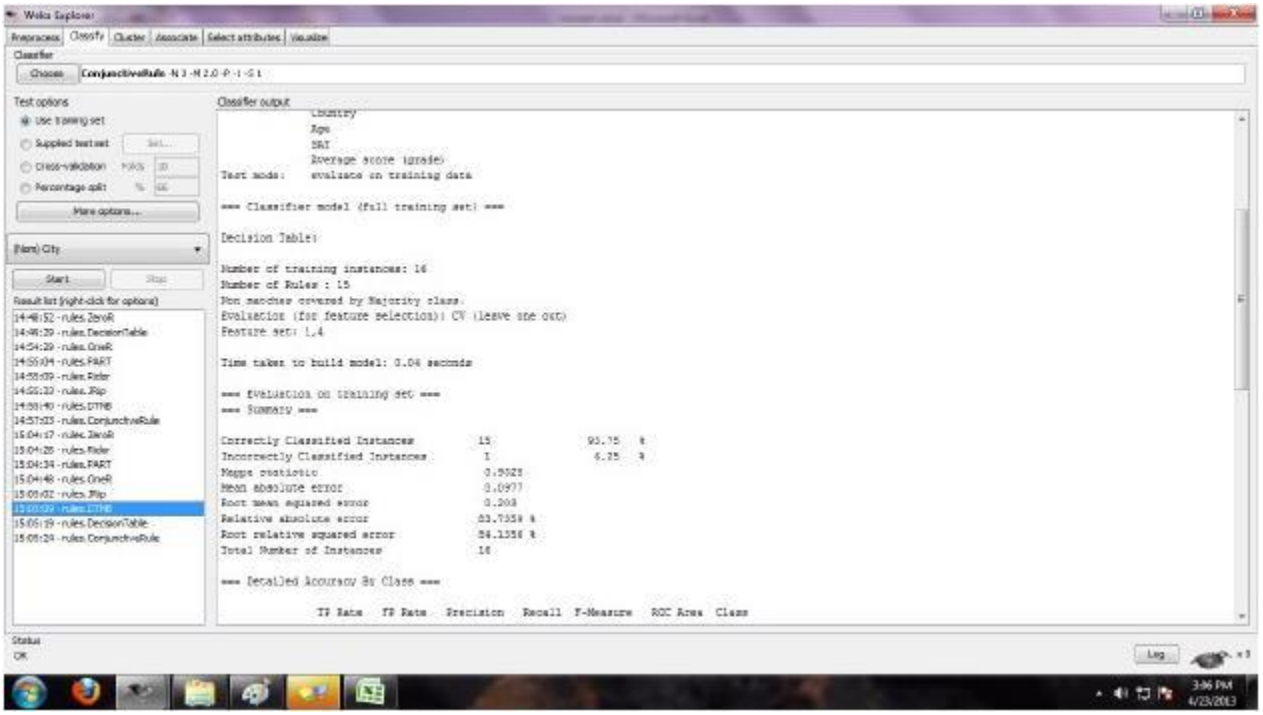
OneR:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) City
Start Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:49:39 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:39 - rules.Ridor
14:56:22 - rules.JRip
14:56:40 - rules.DTNB
14:57:05 - rules.ConjunctiveRule
15:04:17 - rules.ZeroR
15:04:25 - rules.Ridor
15:04:34 - rules.PART
15:04:48 - rules.OneR
15:05:02 - rules.JRip
15:05:39 - rules.DTNB
15:06:19 - rules.DecisionTable
15:06:24 - rules.ConjunctiveRule
Classifier output
DOE24 -> Montreal
DOE25 -> New York
DOE26 -> Hot Coffe
DOE27 -> Java
DOE28 -> Varna
DOE29 -> Moscow
DOE30 -> Drunkard Creek
DOE31 -> Mexican Hat
DOE32 -> Amsterdam
DOE33 -> Mexico
DOE34 -> Elmira
DOE35 -> Lackawana
(16/16 instances correct)
Time taken to build model: 0 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 16 100 %
Incorrectly Classified Instances 0 0 %
Kappa statistic 1
Mean absolute error 0
Root mean squared error 0
Relative absolute error 0 %
Root relative squared error 0 %
Total Number of Instances 16
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
Status
OK
Log
x 0
3:06 PM
4/23/2013

JRip:


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose ConjunctiveRule -N 3 -M 2.0 -P -1 -S 1
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66
More options...
(Nom) City
Start Stop
Result list (right-click for options)
14:48:52 - rules.ZeroR
14:49:39 - rules.DecisionTable
14:54:29 - rules.OneR
14:55:04 - rules.PART
14:55:39 - rules.Ridor
14:56:22 - rules.JRip
14:56:40 - rules.DTNB
14:57:05 - rules.ConjunctiveRule
15:04:17 - rules.ZeroR
15:04:25 - rules.Ridor
15:04:34 - rules.PART
15:04:48 - rules.OneR
15:05:02 - rules.JRip
15:05:39 - rules.DTNB
15:06:19 - rules.DecisionTable
15:06:24 - rules.ConjunctiveRule
Classifier output
Country
Age
SAT
Average score (grade)
Test mode: evaluate on training data
=== Classifier model (full training set) ===
JRIP rules:
=========
=> City=Tel Aviv (16.0/14.0)
Number of Rules : 1
Time taken to build model: 0.01 seconds
=== Evaluation on training set ===
=== Summary ===
Correctly Classified Instances 2 12.5 %
Incorrectly Classified Instances 14 87.5 %
Kappa statistic 0
Mean absolute error 0.1182
Root mean squared error 0.2411
Relative absolute error 99.6816 %
Root relative squared error 99.0951 %
Total Number of Instances 16
=== Detailed Accuracy By Class ===
TP Rate FP Rate Precision Recall F-Measure ROC Area Class
Status
OK
Log
x 0
3:06 PM
4/23/2013

DTNB:



Result :

Thus, the good result (by improving the performance) using the training set and testing data set for numerical values is found out.

EX. No: 7 DATA PRE-PROCESSING – DATA FILTERS

Aim:

To perform the data pre-processing by applying filter.

Objectives:

The data collected from public fourms have plenty of noise or missing data. Weka provides filter to replace the missing values and to remove the noisy data. So that the result will be more accurate.

Algorithm:

1. Download a complete data set (numeric) from UCI.
2. Open the data set in Weka tool.
3. Save the data set with missing values.
4. Apply replace missing value filter.
5. Calculate the accuracy using the formula

$$\text{Accuracy}=\sqrt{\sum (\text{old-new})^2}$$

$$\text{Percentage of accuracy}=\frac{\text{Accuracy}}{\sum \text{old value}} \times 100$$

OUTPUT:

Student Details Table: Missing values

Viewer

Relation: weather

No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: play Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy		96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0		TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny			FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast			TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy		91.0	TRUE	no

Student Details Table: Replace Missing values:

Viewer

Relation: weather-weka.filters.unsupervised.attribute.Replace

No.	1: outlook	2: temperature	3: humidity	4: windy	5: **play**
	Nominal	Numeric	Numeric	Nominal	Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	74.8	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	83.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	74.8	83.0	FALSE	yes
10	rainy	75.0	80.0	FALSE	yes
11	sunny	75.0	70.0	TRUE	yes
12	overcast	74.8	83.0	TRUE	yes
13	overcast	81.0	75.0	FALSE	yes
14	rainy	74.8	91.0	TRUE	no

CALCULATION:

Data Location	Old Data	Predicted data	Errors	(Error)2
J2				
J4				
J6				

Result:

Thus the data pre-processing by applying filter is performed.

EX. No: 8 FEATURE SELECTION

AIM:

To find the good results by feature selection.

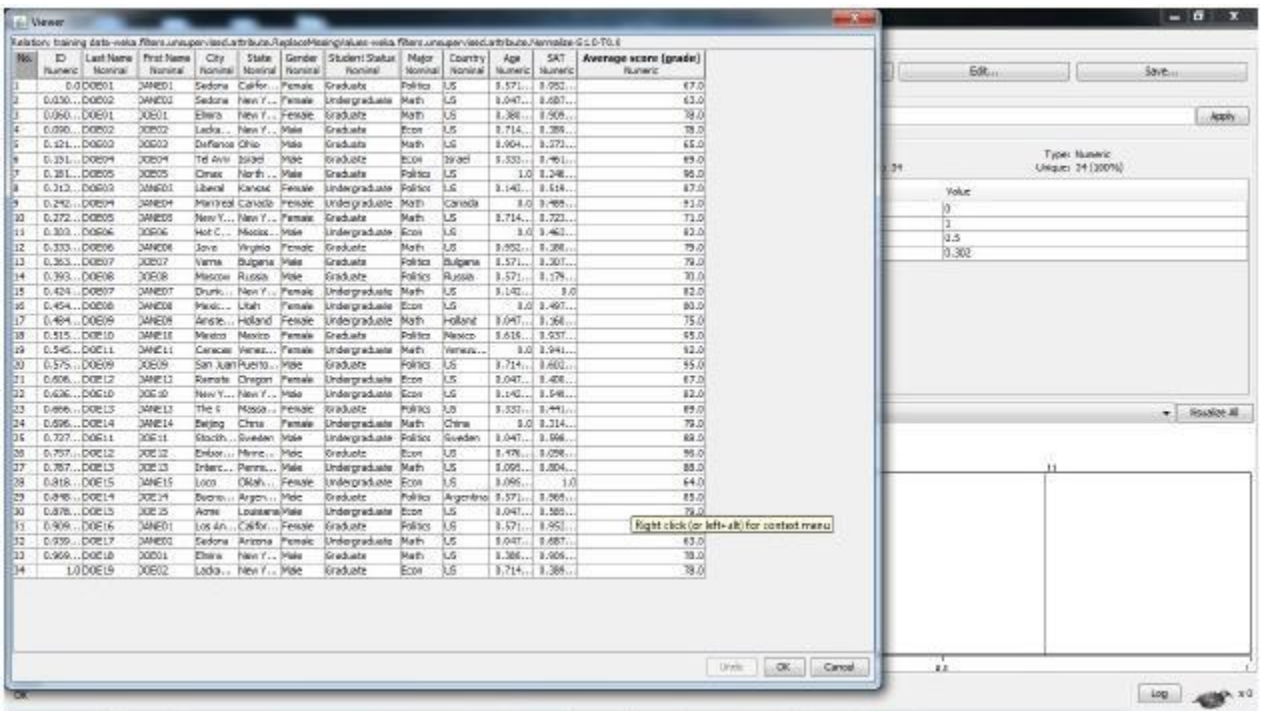
OBJECTIVES:

Any classifier/model has internal feature, those feature gives more accurate and optimal result.

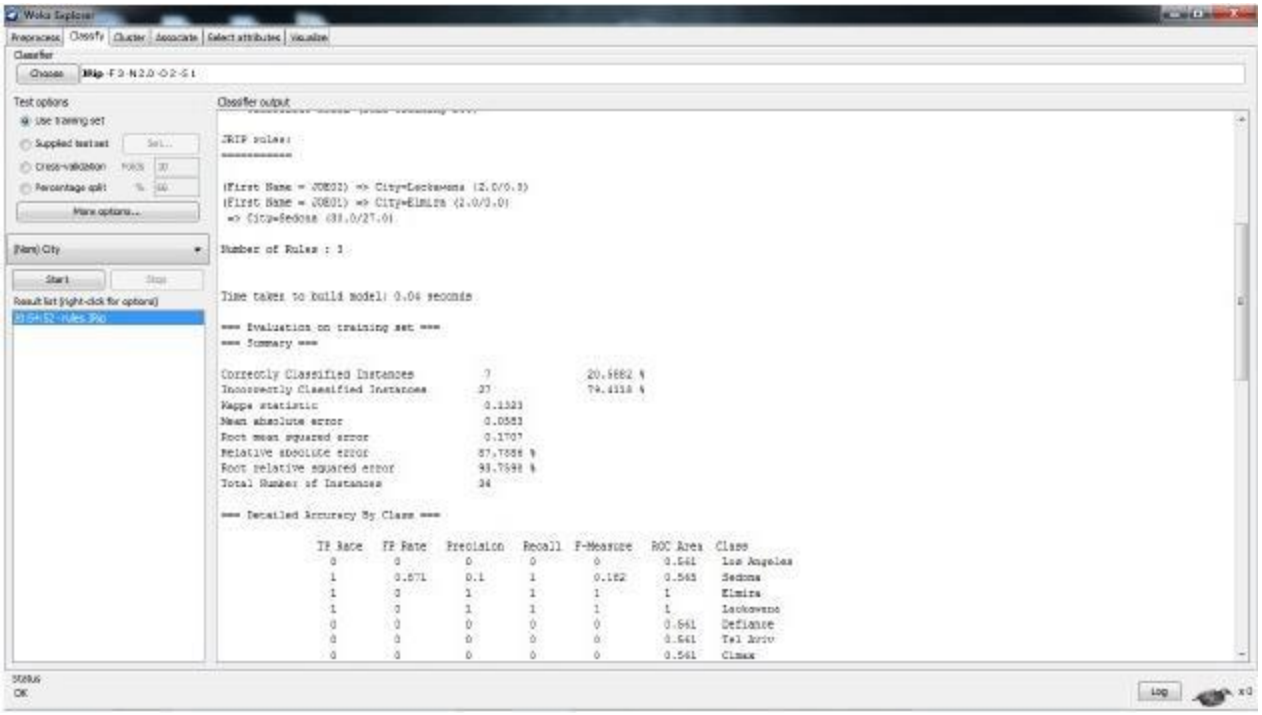
ALGORITHM:

1. Download any dataset with nominal values.
2. Save it as text.arff .
3. Split it into training and testing data set.
4. Go to unsupervised instance remove percentage.
5. Right click on that show properties then select 70% true and save it as training.arff
6. Right click on that show properties then select 70% false and save it as testing.arff using original data set.
7. Open the parameter for classifying .
8. Fix the set of changing values.
9. Look at the performance.
10. Go to step 3 until the expected values of maximum value is reached.

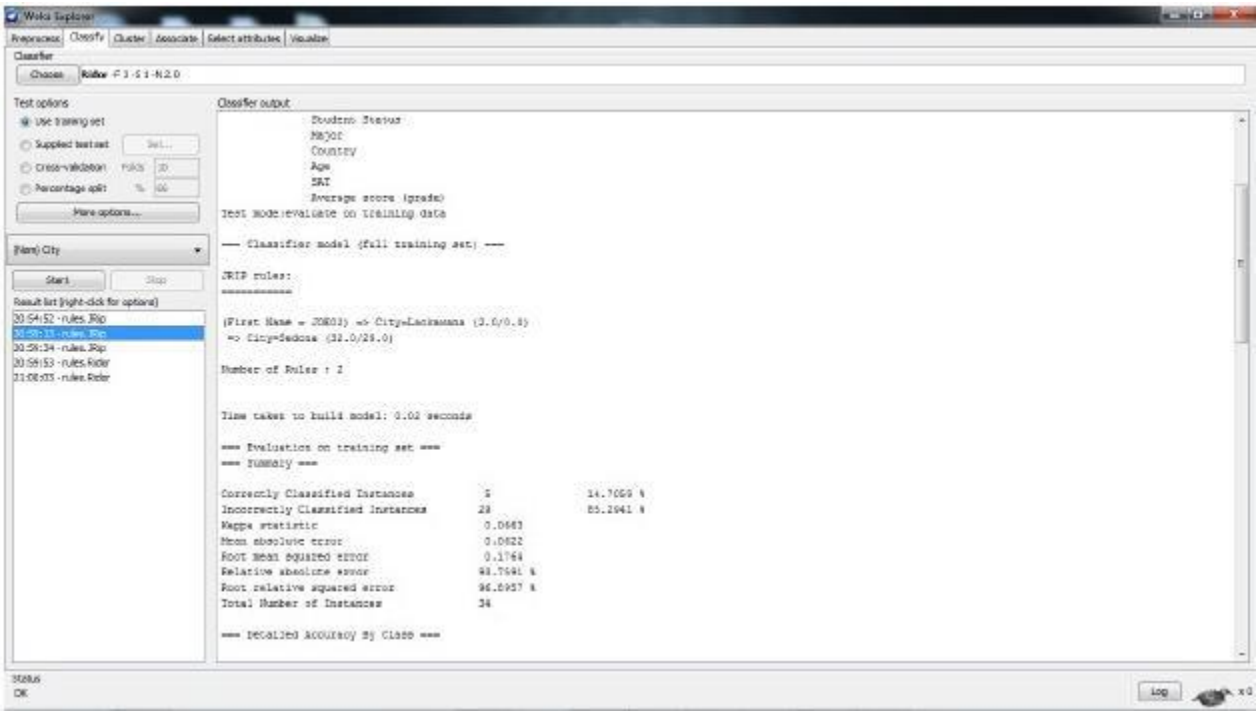
Training Data:



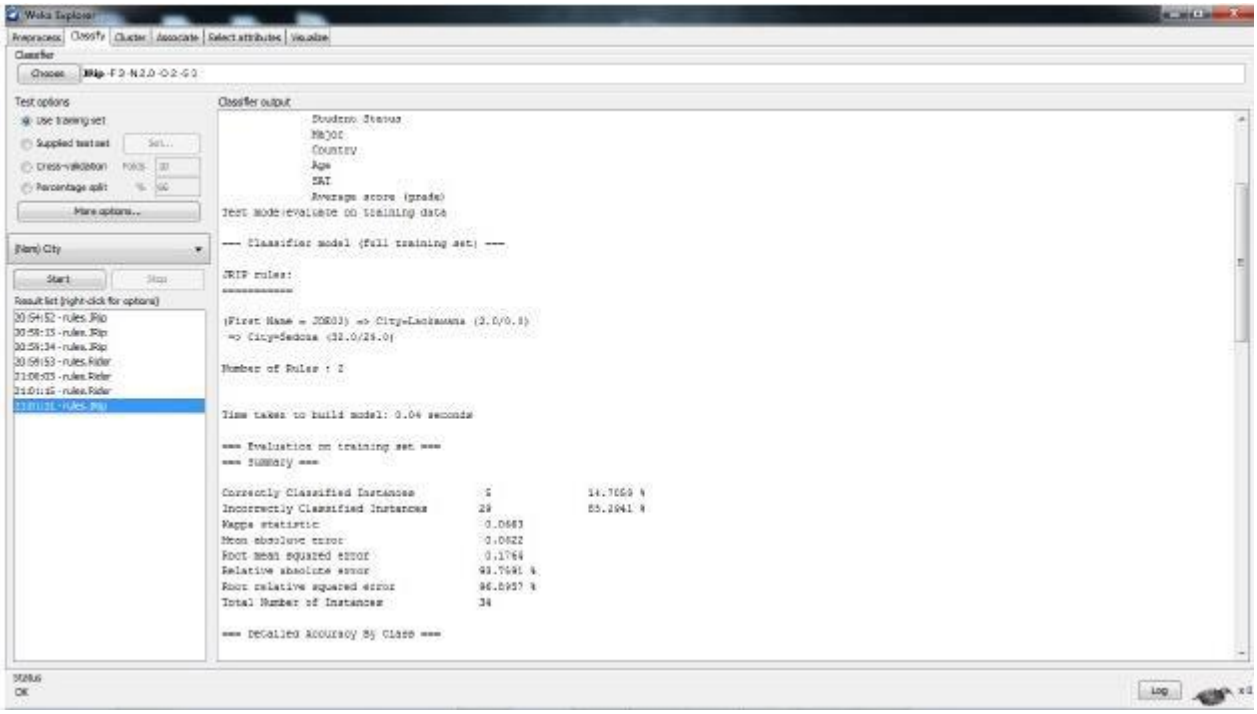
JRip(seed=1):



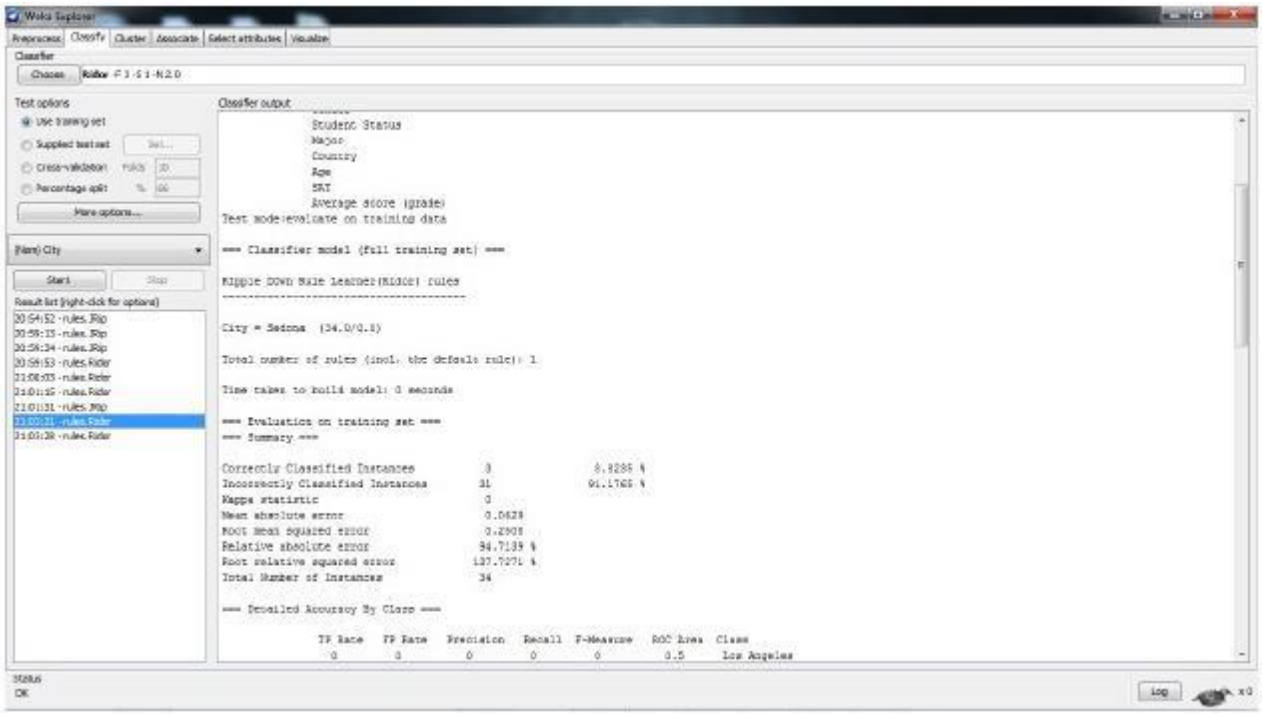
JRip(seed=2):



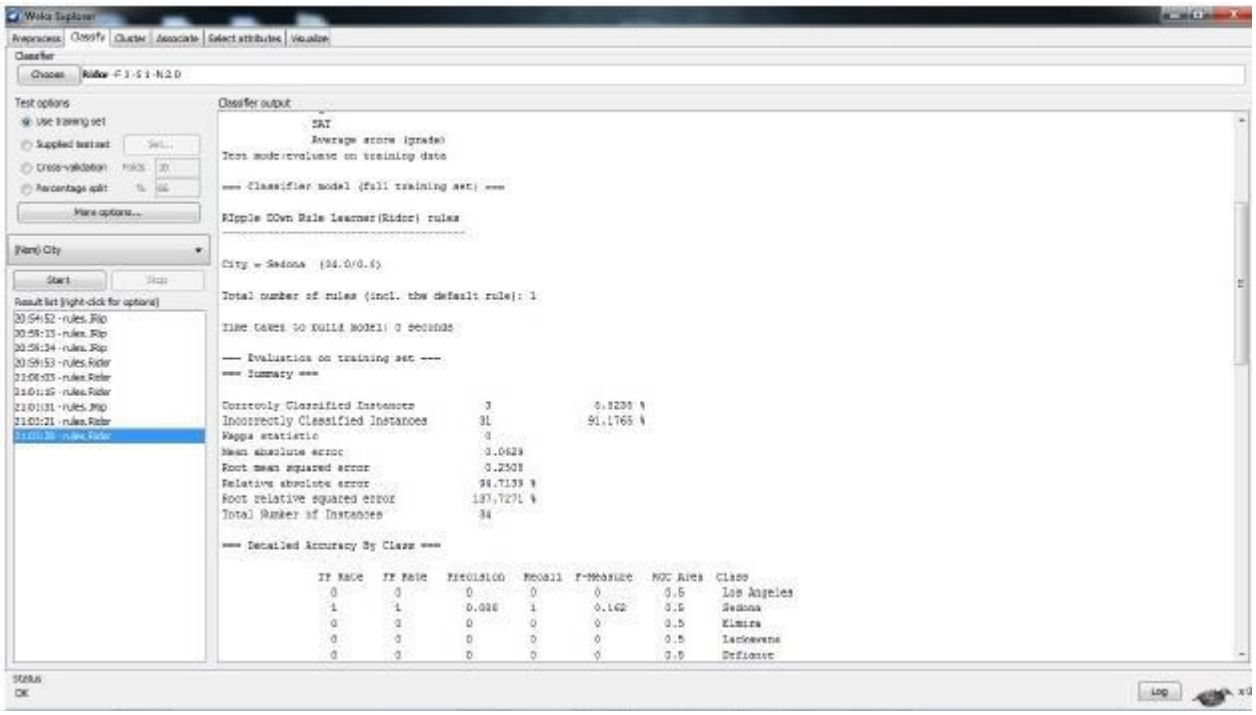
JRip(seed=3):



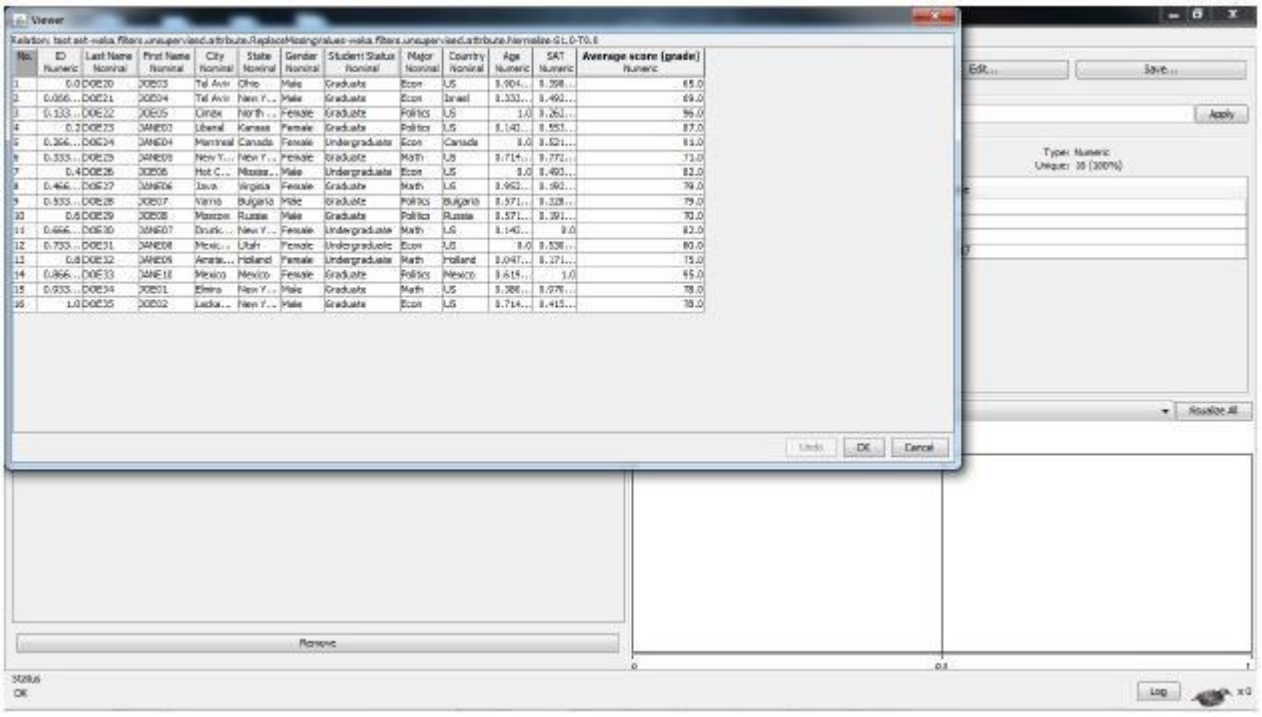
Ridor(seed=1):



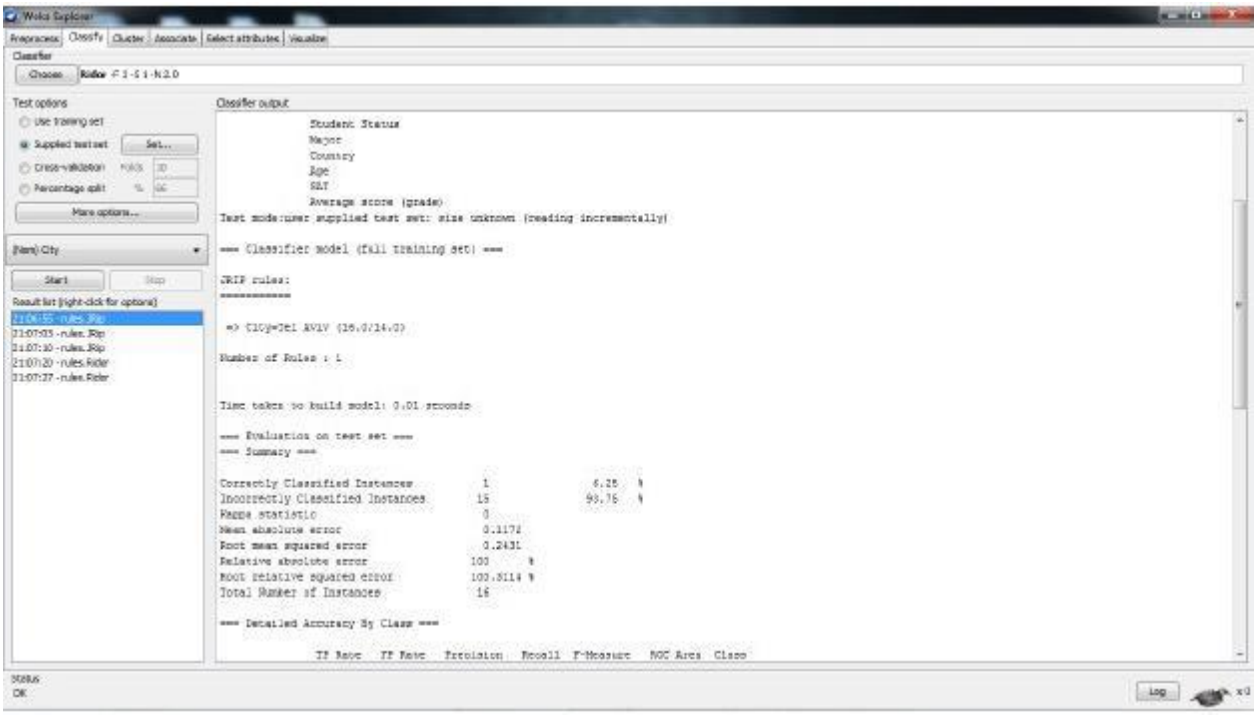
Ridor(seed=2):



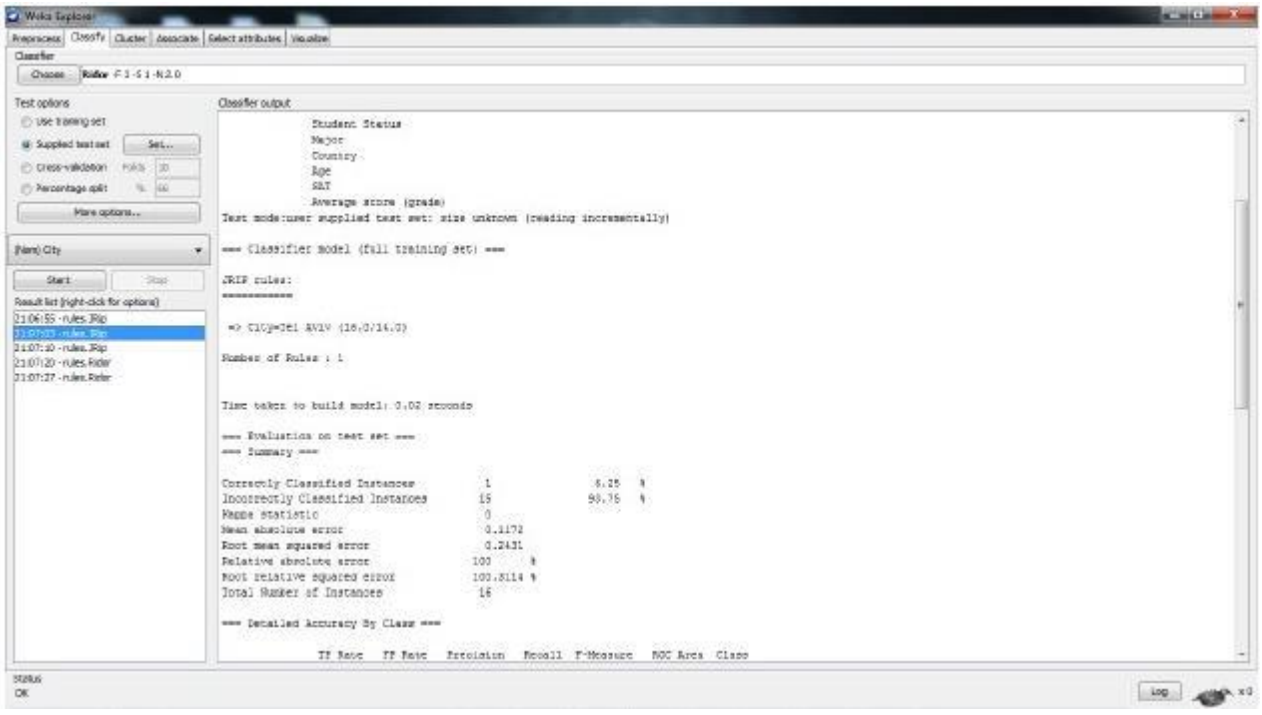
Test Data:



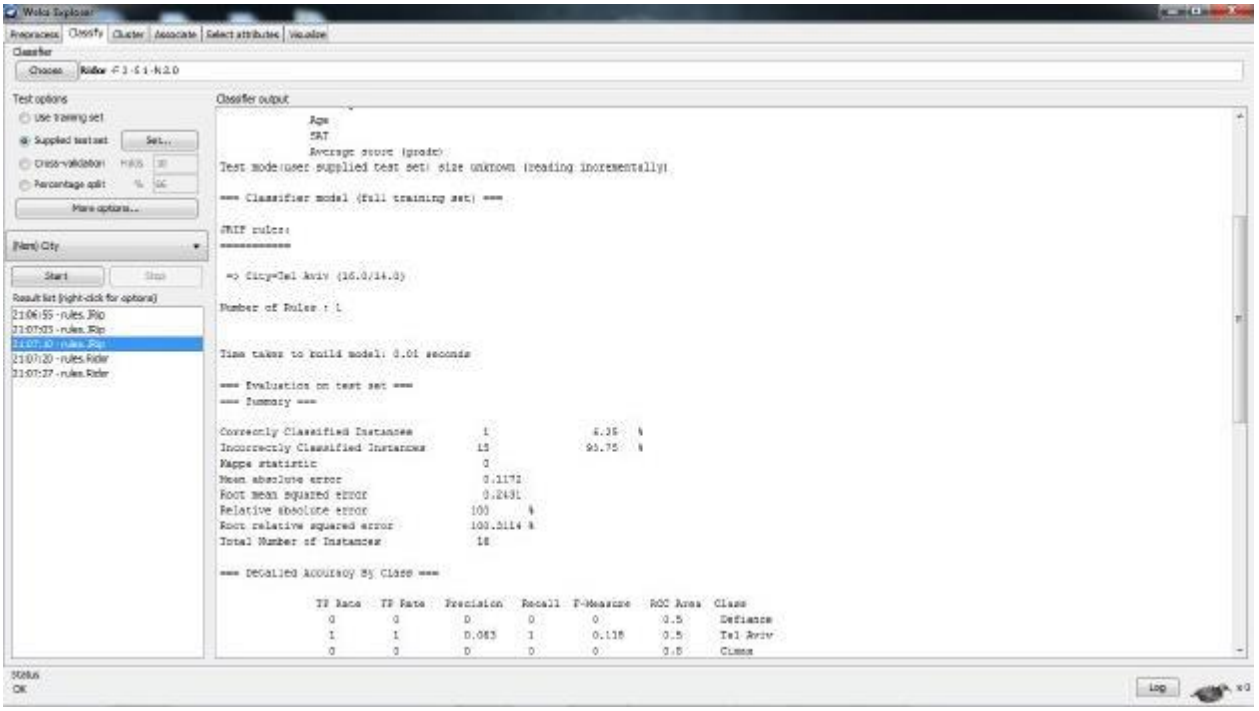
JRip(seed=1):



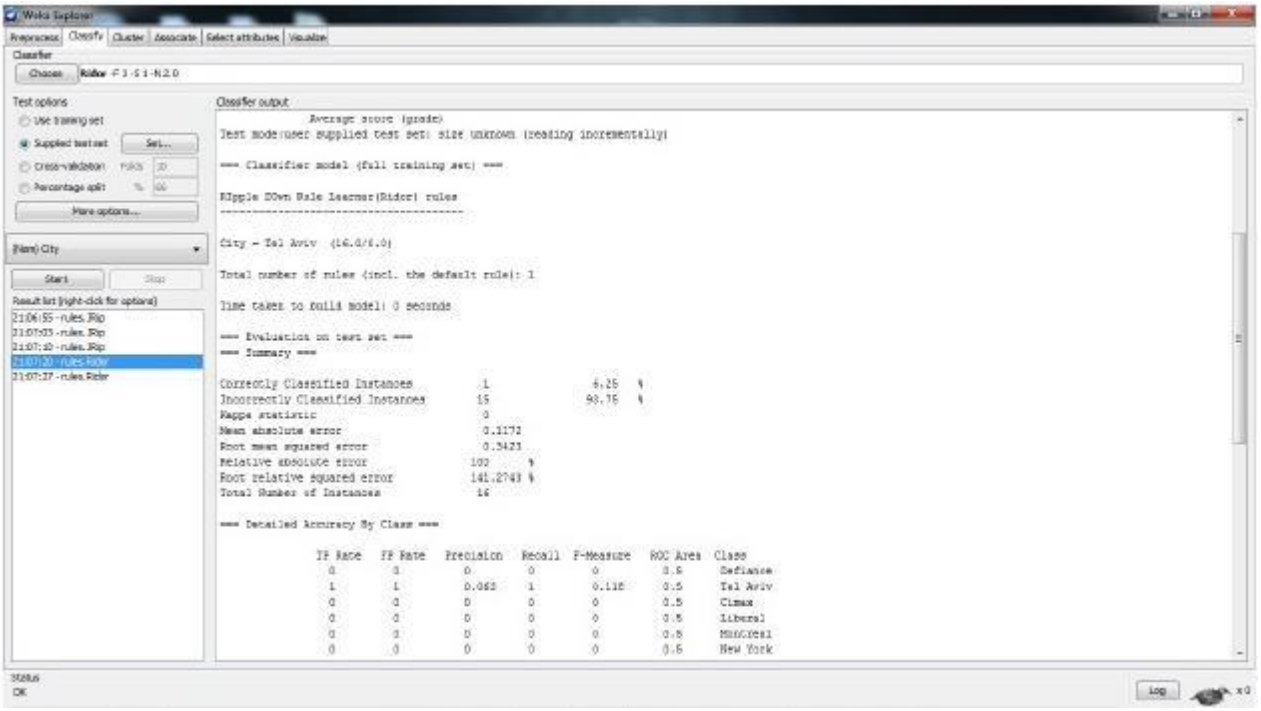
JRip(seed=2):



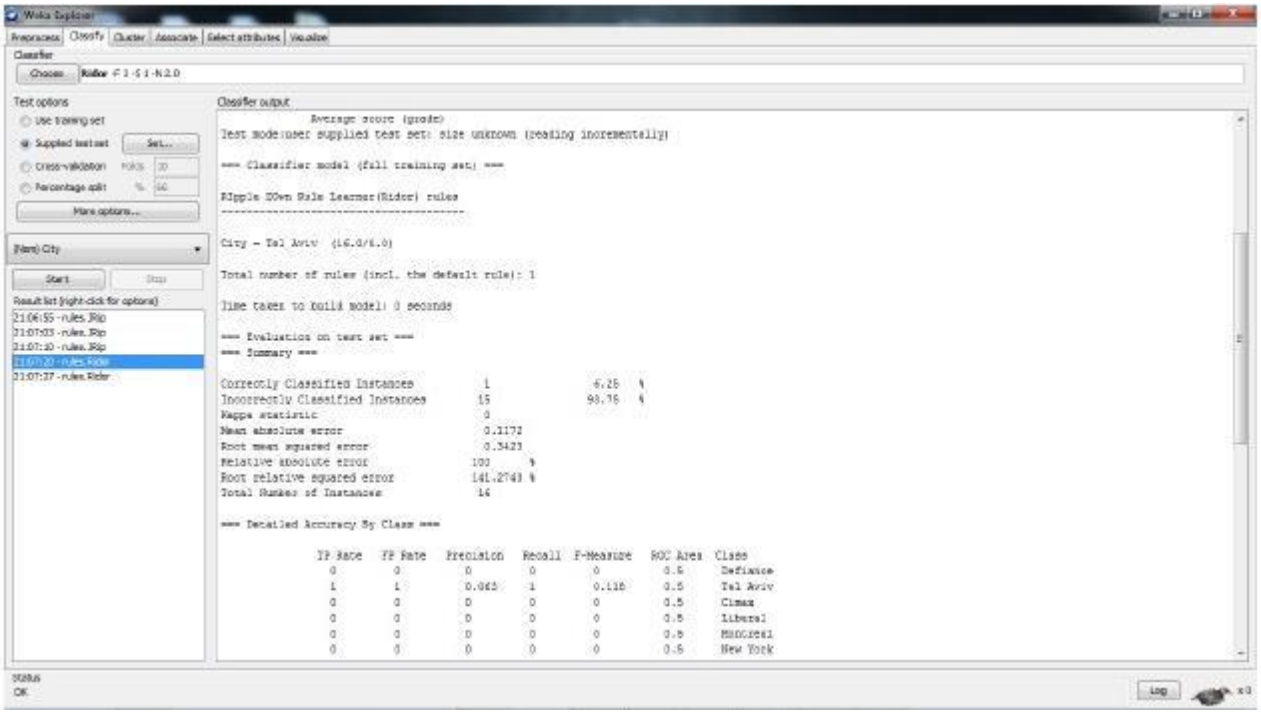
JRip(seed=3):



Ridor(seed=1):



Ridor(seed=2):



Training Data Set Performance:

TRAINING SET		
CLASSIFIER	PARAMETER SETTING	PERFORMANCE
JRip	Seed=1	Root Mean Squared Error=0.1707 Mean Absolute Error=0.0583
JRip	Seed =2	Root Mean Squared Error=0.1764 Mean Absolute Error=0.0622
JRip	Seed =3	Root Mean Squared Error=0.1764 Mean Absolute Error=0.0622
Ridor	Seed =1	Root Mean Squared Error=0.2508 Mean Absolute Error=0.0629
Ridor	Seed=2	Root Mean Squared Error=0.2508 Mean Absolute Error=0.0629

Testing Data set Performance:

TEST SET		
CLASSIFIER	PARAMETER SETTING	PERFORMANCE
JRip	Seed=1	Root Mean Squared Error=0.2431 Mean Absolute Error=0.1172
JRip	Seed =2	Root Mean Squared Error=0.2431 Mean Absolute Error=0.1172
JRip	Seed =3	Root Mean Squared Error=0.2431 Mean Absolute Error=0.1172
Ridor	Seed =1	Root Mean Squared Error=0.3423 Mean Absolute Error=0.1172
Ridor	Seed=2	Root Mean Squared Error=0.3423 Mean Absolute Error=0.1172

Comparison between training and testing data set:

TRAINING		
JRip	Seed=1	Root Mean Squared Error=0.1707 Mean Absolute Error=0.0583
Ridor	Seed =1	Root Mean Squared Error=0.2508 Mean Absolute Error=0.0629

TEST		
JRip	Seed=1	Root Mean Squared Error=0.2431 Mean Absolute Error=0.1172
Rider	Seed =1	Root Mean Squared Error=0.3423 Mean Absolute Error=0.1172

Result:

Thus the good results by feature selection were found.

EX. No: 9 Web Mining

Aim:

To apply the web mining technique clustering algorithm for the given dataset.

Introduction to Web Mining:

Web mining is an application of data mining techniques to find information patterns from the web data. Web mining helps to improve the power of web search engine by identifying the web pages and classifying the web documents.Web mining is very useful to e-commerce websites and e-services.

Web Content Mining :

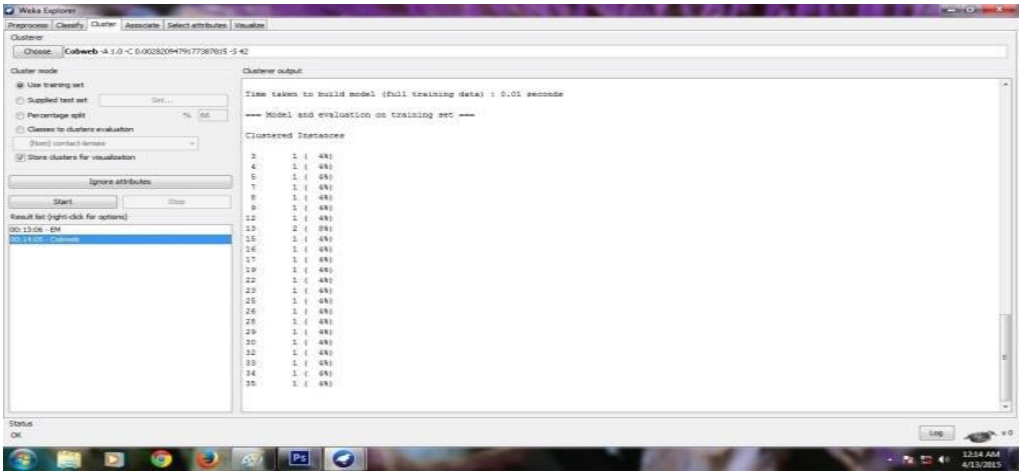
Web content mining can be used for mining of useful data, information and knowledge from web page content. Web structure mining helps to find useful knowledge or information pattern from the structure of hyperlinks. Due to heterogeneity and absence of structure in web data, automated discovery of new knowledge pattern can be challenging to some extent.Web content mining performs scanning and mining of the text, images and groups of web pages according to the content of the input (query), by displaying the list in search engines. For example: If an user wants to search for a particular book, then search engine provides the list of suggestions.

ALGORITHM:

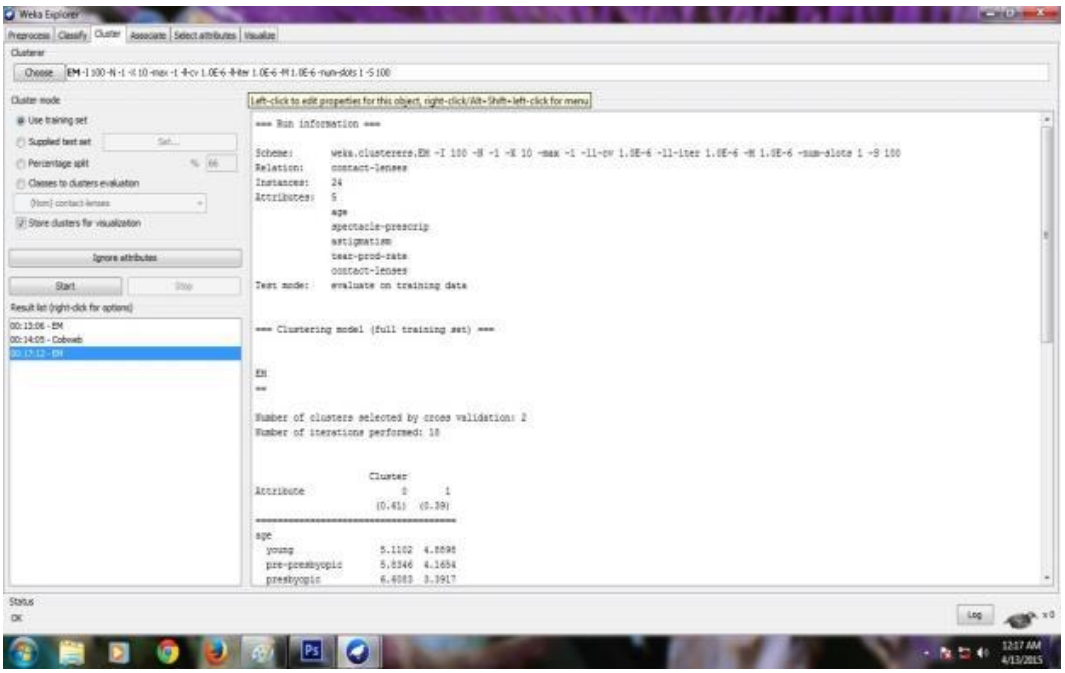
1. Open the weka tool.
2. Download a dataset by using UCI.
3. Apply replace missing values.
4. Apply normalize filter.
5. Click the cluster tab.
6. Apply all algorithms one by one.
7. Find the no of clusters that are formed
8. Note the output.

Output:

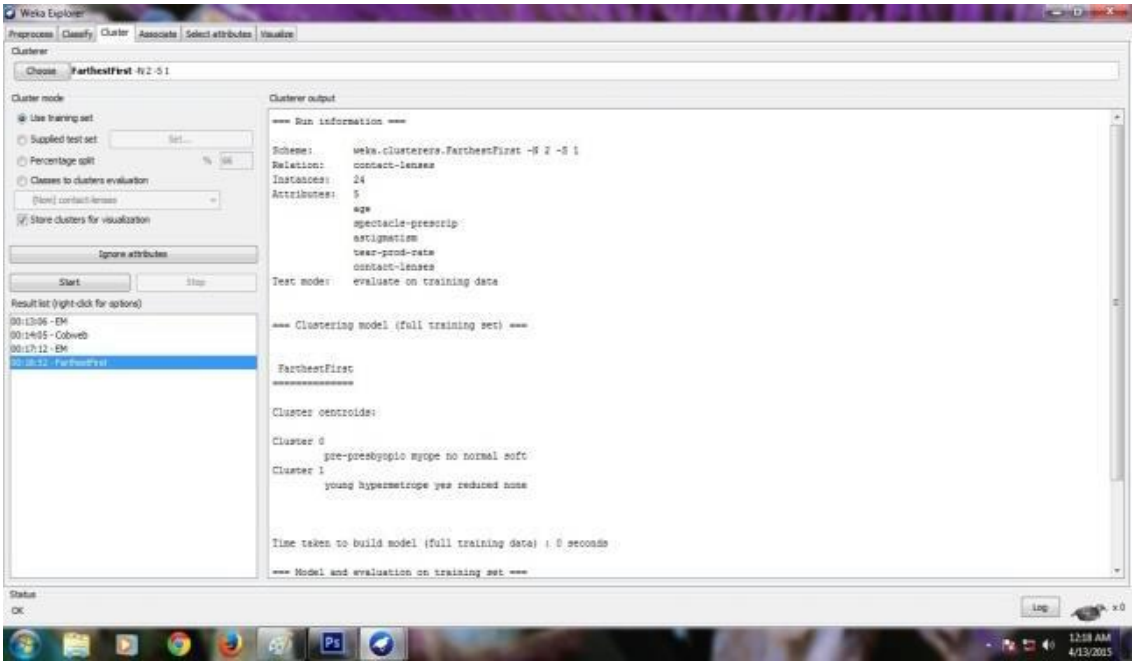
Cobweb



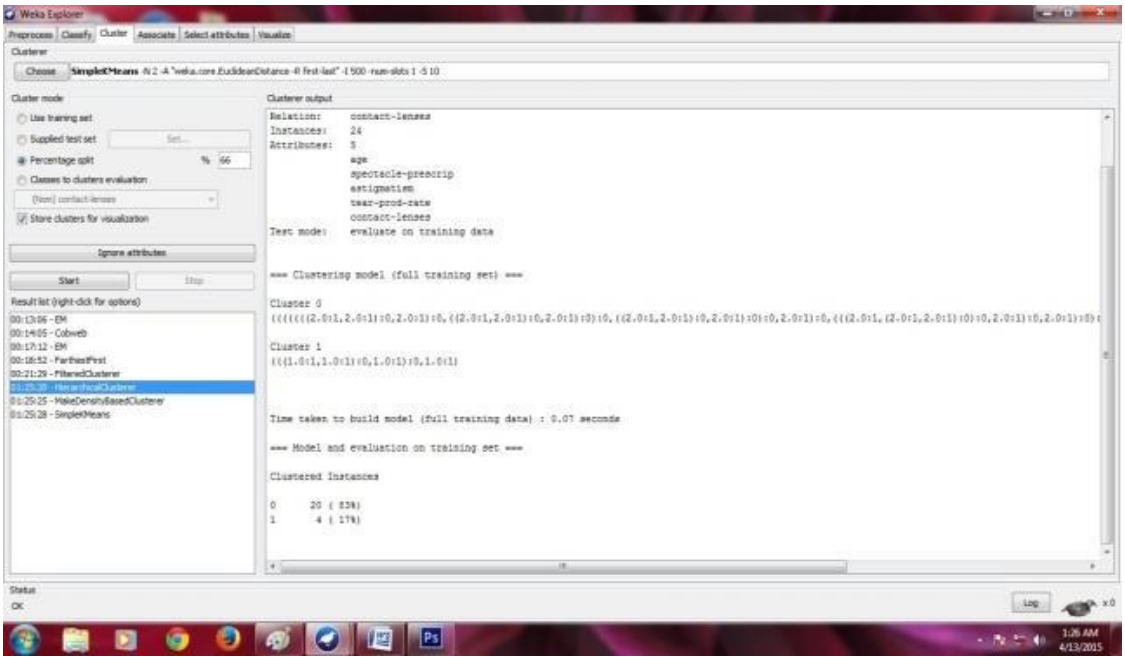
EM



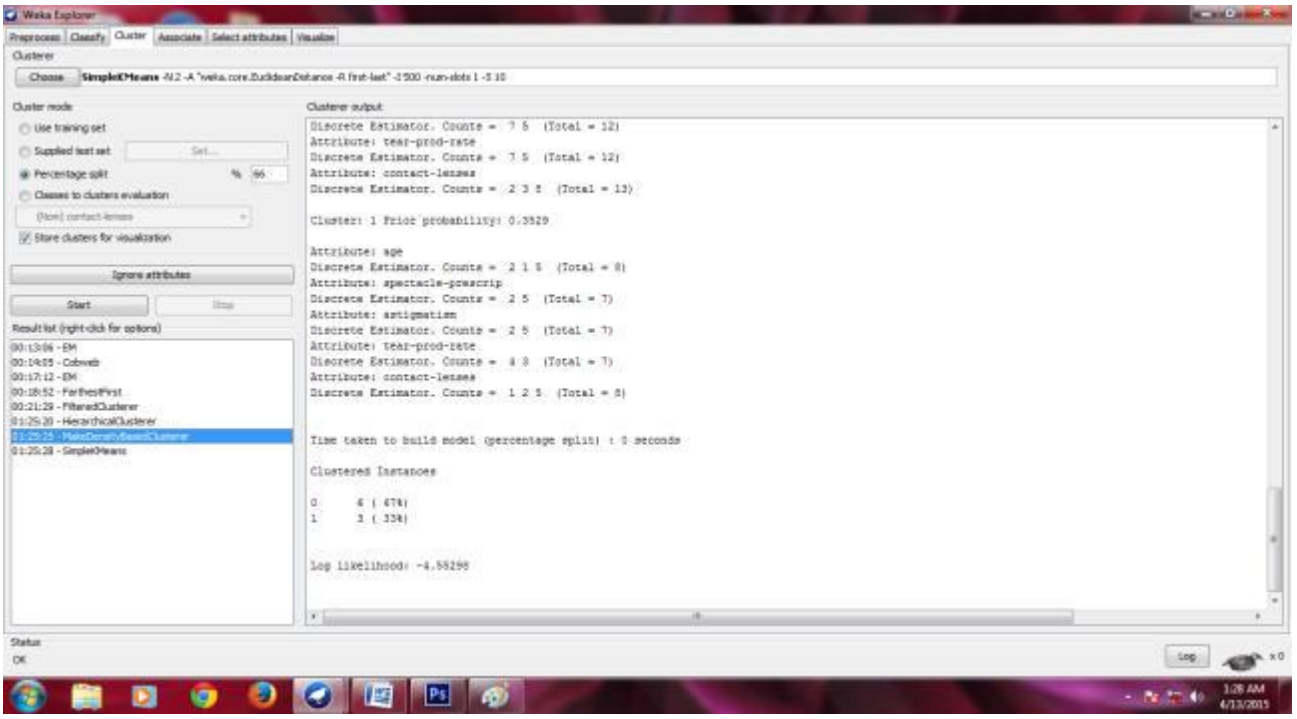
Farthest First



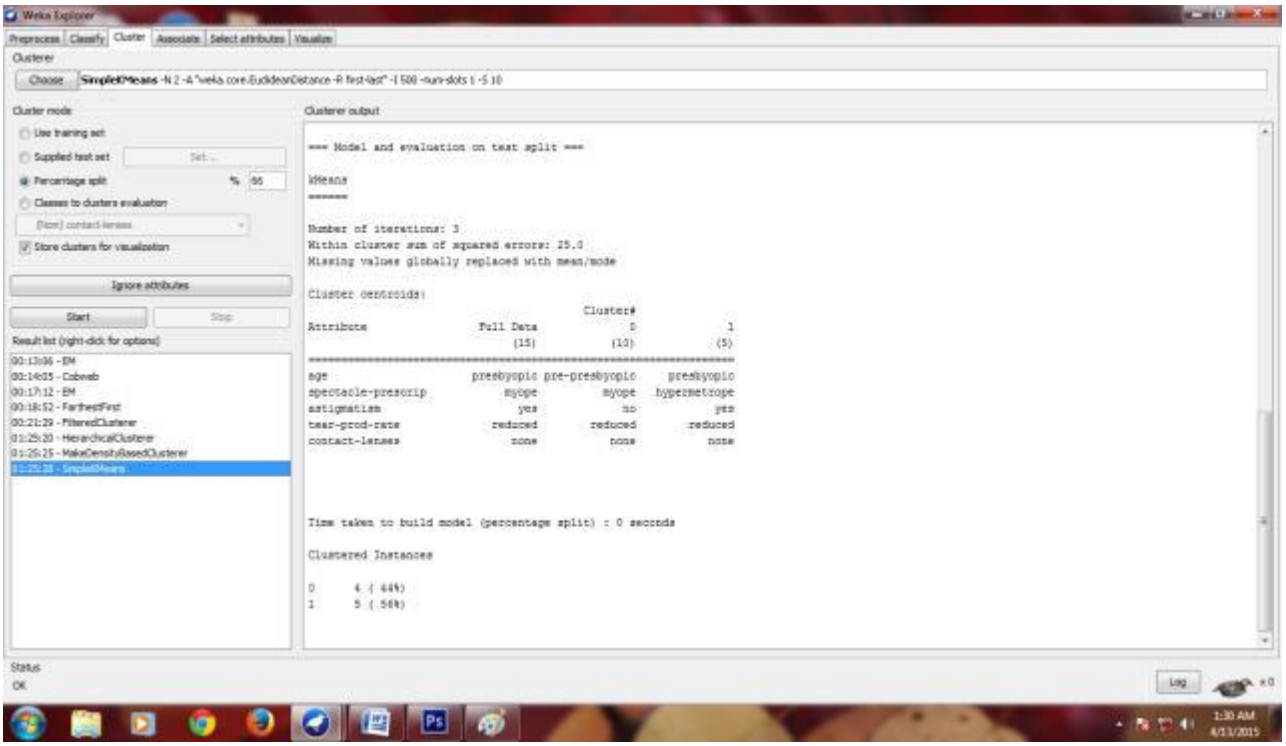
Filtered Cluster:



Hierarchical Cluster



Simple KMeans:



Result:

Thus the web mining technique clustering algorithm for the given dataset is implemented.

EX. No: 10 **TEXT MINING**

Aim:

To find association between data and to find the frequent item set for text mining.

Text Data Mining

Text data mining can be described as the process of extracting essential data from standard language text. All the data that we generate via text messages, documents, emails, files are written in common language text. Text mining is primarily used to draw useful insights or patterns from such data.The purchasing of one product when another product is purchased represents an association rule. Association rules are frequently used by retail store to assist in marketing, advertising, floor placement, and inventory control. Association rules are used to show the relationship between data items.

Keyword-based Association Analysis in text mining:

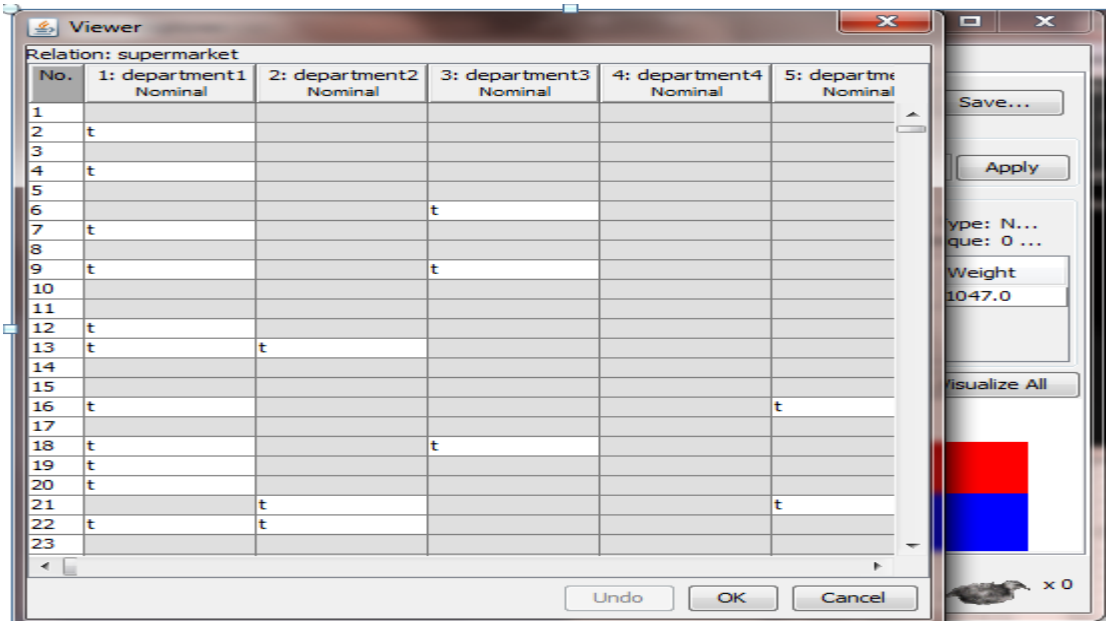
It collects sets of keywords or terms that often happen together and afterward discover the association relationship among them. First, it preprocesses the text data by parsing, stemming, removing stop words, etc. Once it pre-processed the data, then it induces association mining algorithms. Here, human effort is not required, so the number of unwanted results and the execution time is reduced.

ALGORITHM:

1. Open dataset
2. Select associate
3. Choose different algorithm for association
4. Observe the performance
5. Select the association rule with the maximum confidence rule.

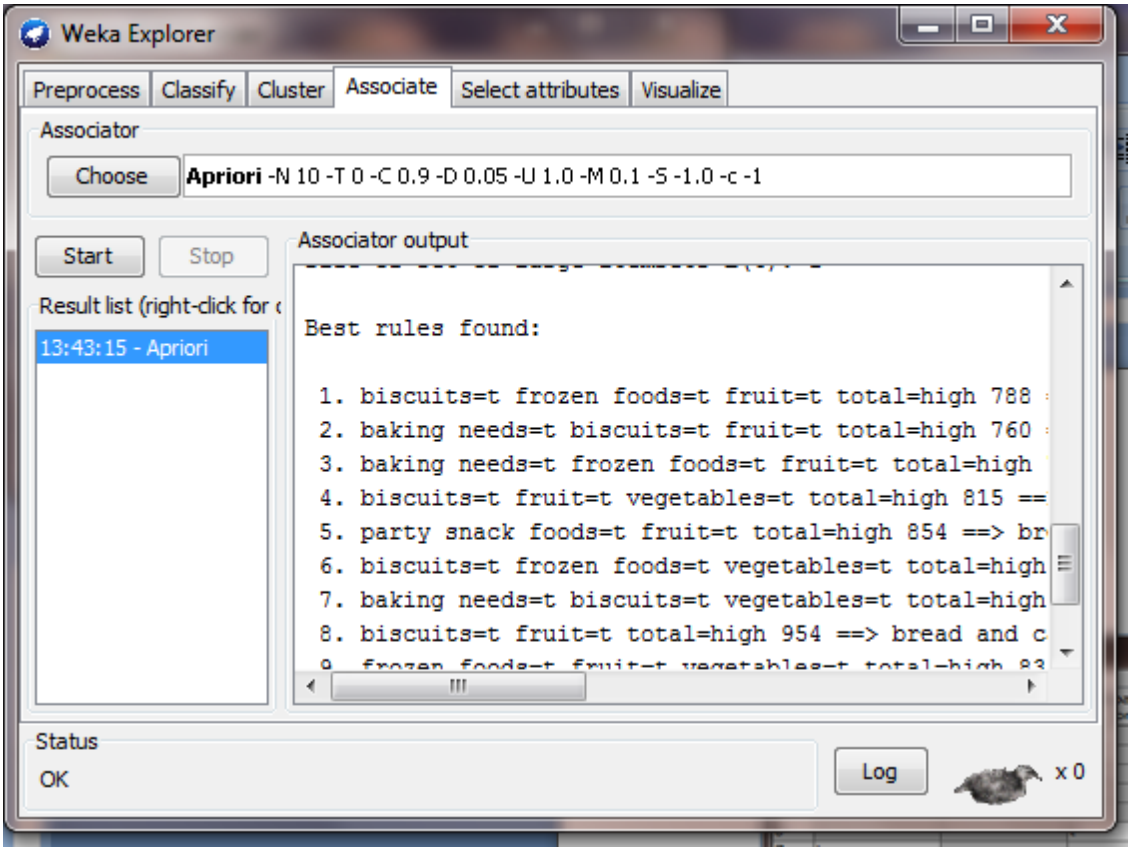
INPUT:

SuperMarket data set

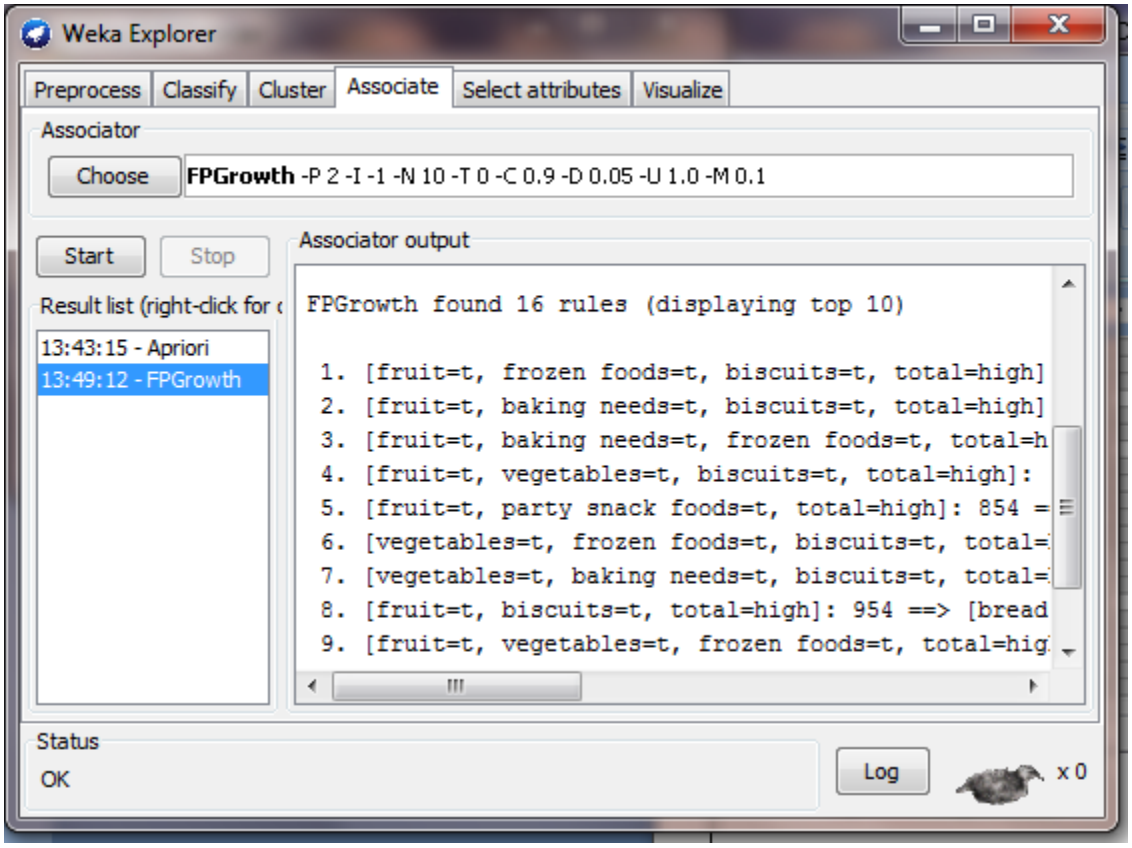


OUTPUT:

Apriori Algorithm



FP-Growth Algorithm:



Result:

Thus association between data and to find the frequent item set for text mining was found.

EX. No: 11 DESIGN OF FACT AND DIMENSION TABLES

Aim:

To design fact and dimension tables.

Fact Table :

A fact table is used in the dimensional model in data warehouse design. A fact table is found at the center of a star schema or snowflake schema surrounded by dimension tables.A fact table consists of facts of a particular business process e.g., sales revenue by month by product. Facts are also known as measurements or metrics. A fact table record captures a measurement or a metric.

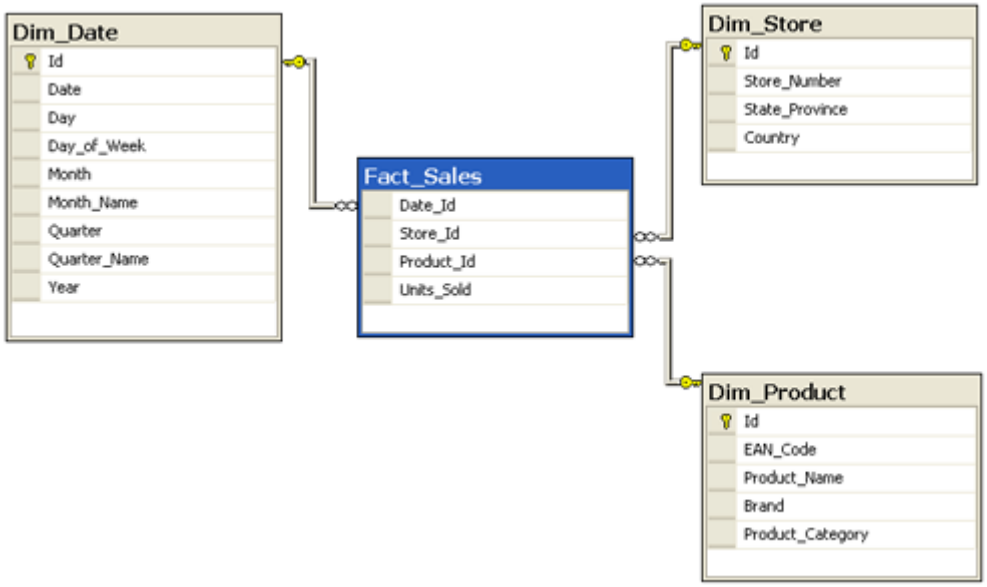
Designing fact table steps

Here is overview of four steps to designing a fact table:

1. **Choosing business process to model** – The first step is to decide what business process to model by gathering and understanding business needs and available data
2. **Declare the grain** – by declaring a grain means describing exactly what a fact table record represents
3. **Choose the dimensions** – once grain of fact table is stated clearly, it is time to determine dimensions for the fact table.
4. **Identify facts** – identify carefully which facts will appear in the fact table.

Fact table FACT_SALES that has a grain which gives us a number of units sold by date, by store and by product.

All other tables such as DIM_DATE, DIM_STORE and DIM_PRODUCT are dimensions tables. This schema is known as the star schema.



Result: Thus design fact and dimension tables are created.

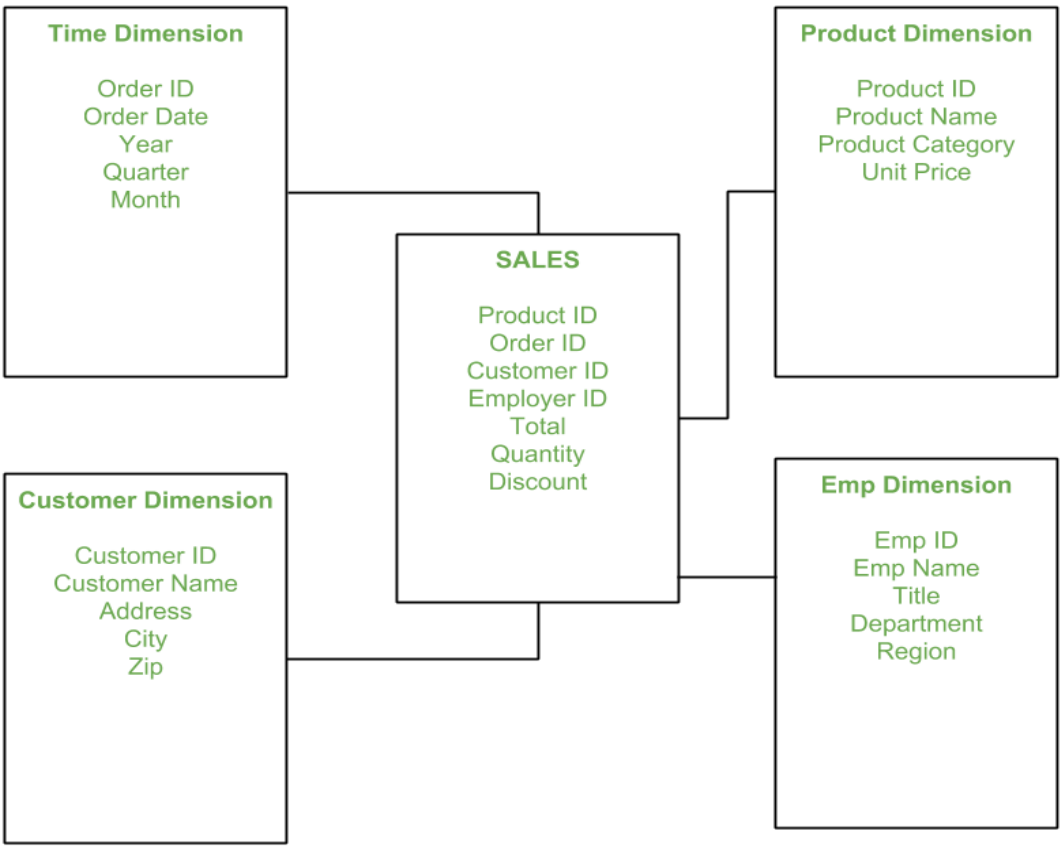
EX. No: 12 GENERATING GRAPHS FOR STAR SCHEMA

Aim:

To generate graphs for star schema.

Introduction:

Star schema is the fundamental schema among the data mart schema and it is simplest. This schema is widely used to develop or build a data warehouse and dimensional data marts. It includes one or more fact tables indexing any number of dimensional tables. The star schema is a necessary case of the snowflake schema. It is also efficient for handling basic queries.It is said to be star as its physical model resembles to the star shape having a fact table at its center and the dimension tables at its peripheral representing the star's points.



In the above demonstration, SALES is a fact table having attributes i.e. (Product ID, Order ID, Customer ID, Employer ID, Total, Quantity, Discount) which references to the dimension tables. **Employee dimension table** contains the attributes: Emp ID, Emp Name, Title, Department and Region. Product dimension table contains the attributes: Product ID, Product Name, Product Category, Unit Price. Customer dimension table contains the attributes: Customer ID, Customer Name, Address, City, Zip. Time dimension table contains the attributes: Order ID, Order Date, Year, Quarter, Month.

In Star Schema, Business process data, that holds the quantitative data about a business is distributed in fact tables, and dimensions which are descriptive characteristics related to fact data. Sales price, sale quantity, distant, speed, weight, and weight measurements are few examples of fact data in star schema. Often, A Star Schema having multiple dimensions is termed as Centipede Schema. It is easy to handle a star schema which has dimensions of few attributes.

Result: Thus the graphs for star schema are generated.