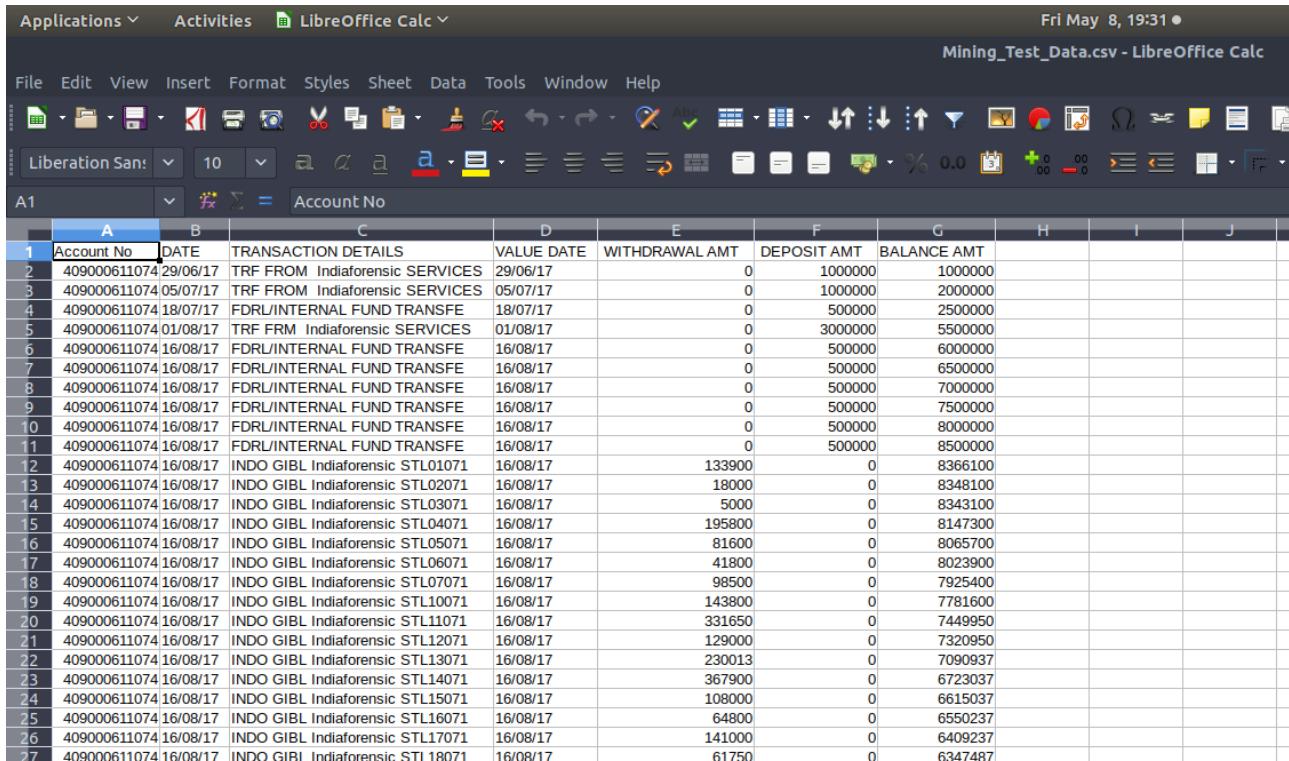


DATA WAREHOUSE AND DATA MINING PRACTICAL FILE

**Submitted by :
Shiv Kumar
2016UIT2563
IT-2
8th Semester**

Practical 1 : Create an Employee Table with the help of Data Mining Tool WEKA

1. Create an employee table with required entries in the MS-Excel (spreadsheet).

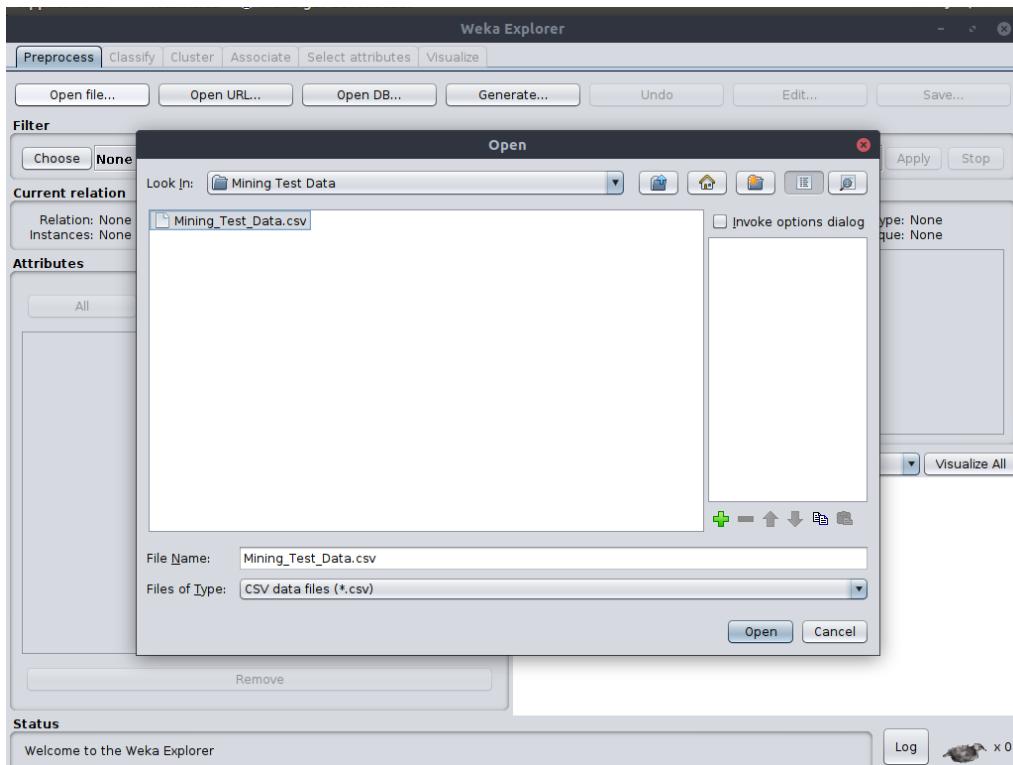


The screenshot shows a LibreOffice Calc spreadsheet window titled "Mining_Test_Data.csv - LibreOffice Calc". The data consists of 27 rows of transaction details. The columns are labeled A through J, with headers including "Account No", "DATE", "TRANSACTION DETAILS", "VALUE DATE", "WITHDRAWAL AMT", "DEPOSIT AMT", and "BALANCE AMT". The data includes various transactions such as TRF FROM Indiaforensic SERVICES, FDRL/INTERNAL FUND TRANSFE, and INDO GIBL Indiaforensic STL01071.

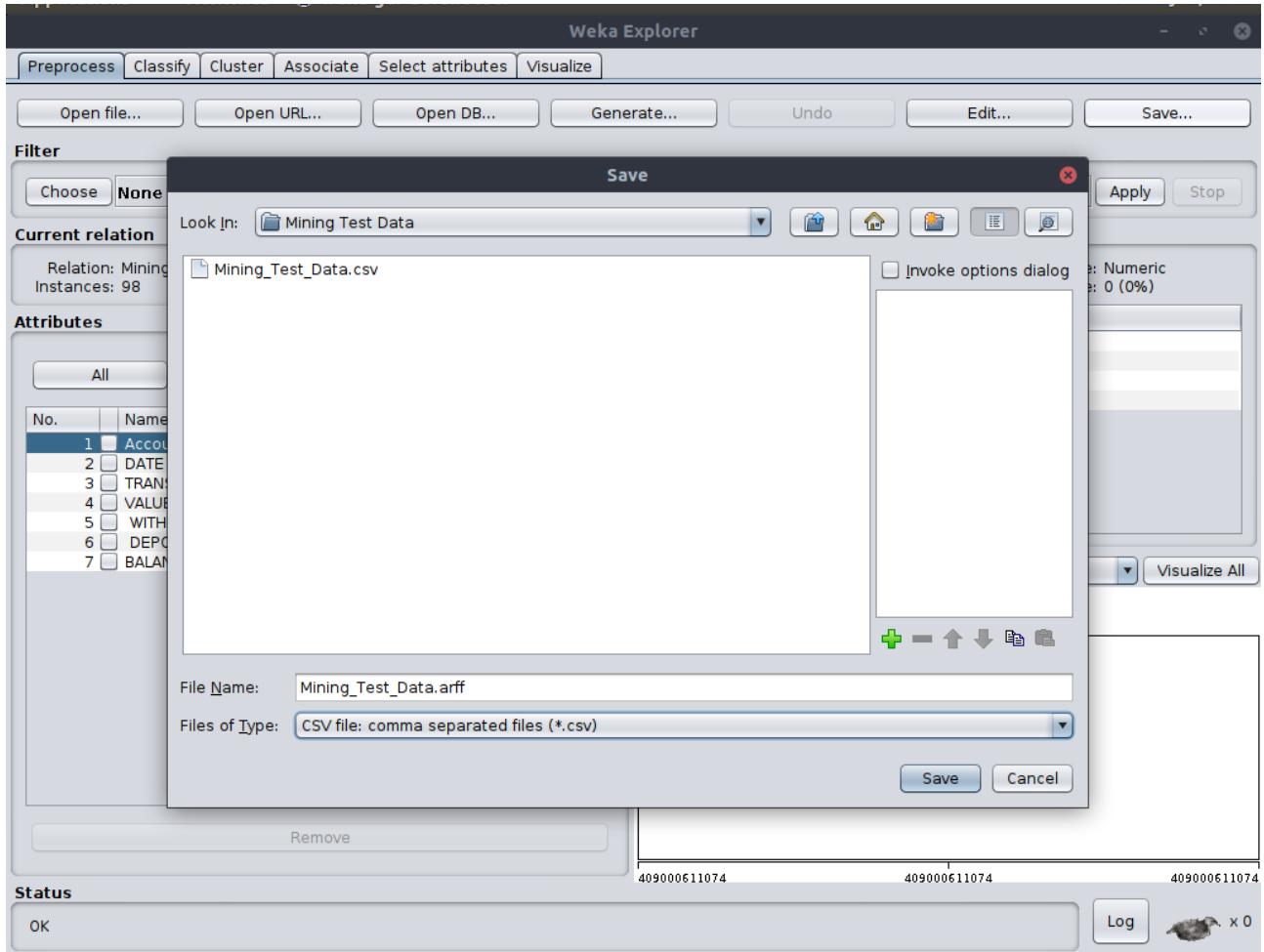
A	B	C	D	E	F	G	H	I	J
1	Account No	DATE	TRANSACTION DETAILS	VALUE DATE	WITHDRAWAL AMT	DEPOSIT AMT	BALANCE AMT		
2	409000611074	29/06/17	TRF FROM Indiaforensic SERVICES	29/06/17	0	1000000	1000000		
3	409000611074	05/07/17	TRF FROM Indiaforensic SERVICES	05/07/17	0	1000000	2000000		
4	409000611074	18/07/17	FDRL/INTERNAL FUND TRANSFE	18/07/17	0	500000	2500000		
5	409000611074	01/08/17	TRF FRM Indiaforensic SERVICES	01/08/17	0	300000	5500000		
6	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	6000000		
7	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	6500000		
8	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	7000000		
9	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	7500000		
10	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	8000000		
11	409000611074	16/08/17	FDRL/INTERNAL FUND TRANSFE	16/08/17	0	500000	8500000		
12	409000611074	16/08/17	INDO GIBL Indiaforensic STL01071	16/08/17	133900	0	8366100		
13	409000611074	16/08/17	INDO GIBL Indiaforensic STL02071	16/08/17	18000	0	8348100		
14	409000611074	16/08/17	INDO GIBL Indiaforensic STL03071	16/08/17	5000	0	8343100		
15	409000611074	16/08/17	INDO GIBL Indiaforensic STL04071	16/08/17	195800	0	8147300		
16	409000611074	16/08/17	INDO GIBL Indiaforensic STL05071	16/08/17	81600	0	8065700		
17	409000611074	16/08/17	INDO GIBL Indiaforensic STL06071	16/08/17	41800	0	8023900		
18	409000611074	16/08/17	INDO GIBL Indiaforensic STL07071	16/08/17	98500	0	7925400		
19	409000611074	16/08/17	INDO GIBL Indiaforensic STL10071	16/08/17	143800	0	7781600		
20	409000611074	16/08/17	INDO GIBL Indiaforensic STL11071	16/08/17	331650	0	7449950		
21	409000611074	16/08/17	INDO GIBL Indiaforensic STL12071	16/08/17	129000	0	7320950		
22	409000611074	16/08/17	INDO GIBL Indiaforensic STL13071	16/08/17	230013	0	7090937		
23	409000611074	16/08/17	INDO GIBL Indiaforensic STL14071	16/08/17	367900	0	6723037		
24	409000611074	16/08/17	INDO GIBL Indiaforensic STL15071	16/08/17	108000	0	6615037		
25	409000611074	16/08/17	INDO GIBL Indiaforensic STL16071	16/08/17	64800	0	6550237		
26	409000611074	16/08/17	INDO GIBL Indiaforensic STL17071	16/08/17	141000	0	6409237		
27	409000611074	16/08/17	INDO GIRI Indiaforensic STI 18071	16/08/17	61750	0	6347487		

2. Save the file as .csv file from the spreadsheet software

3. Open .csv file in weka mining tool explorer

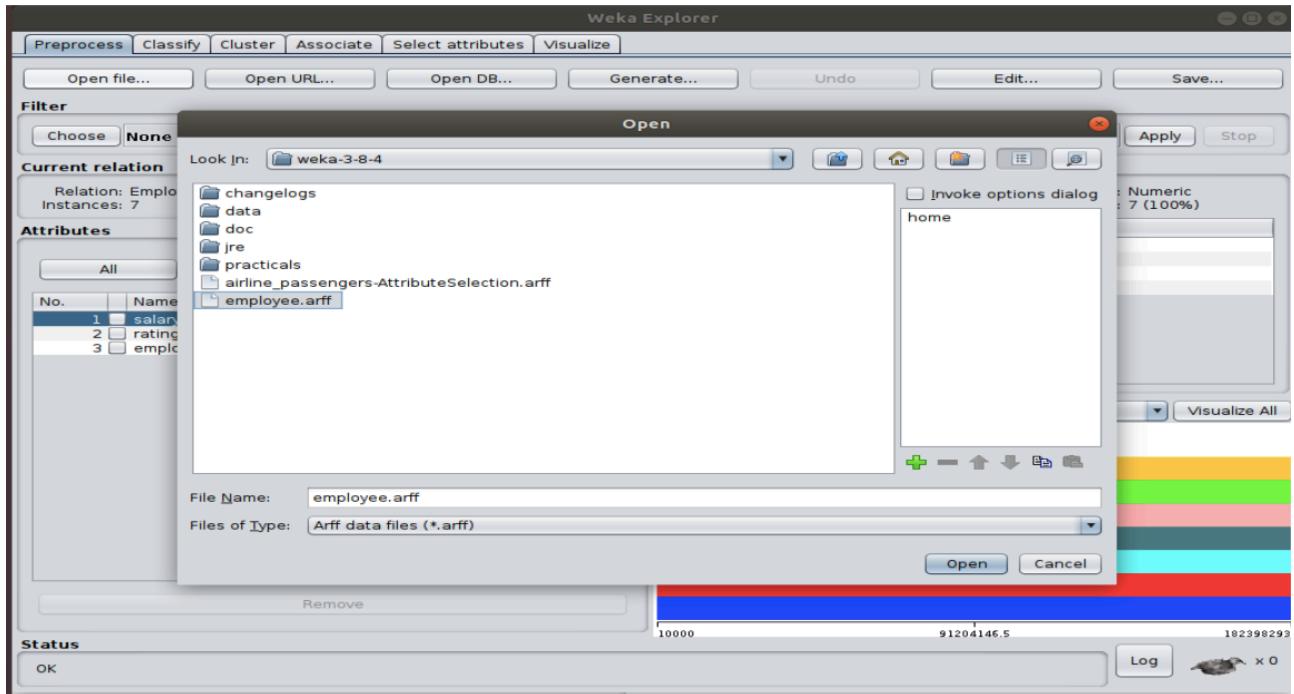


4. Save file as .arff from weka mining tool to create .arff file for the given employee table data.

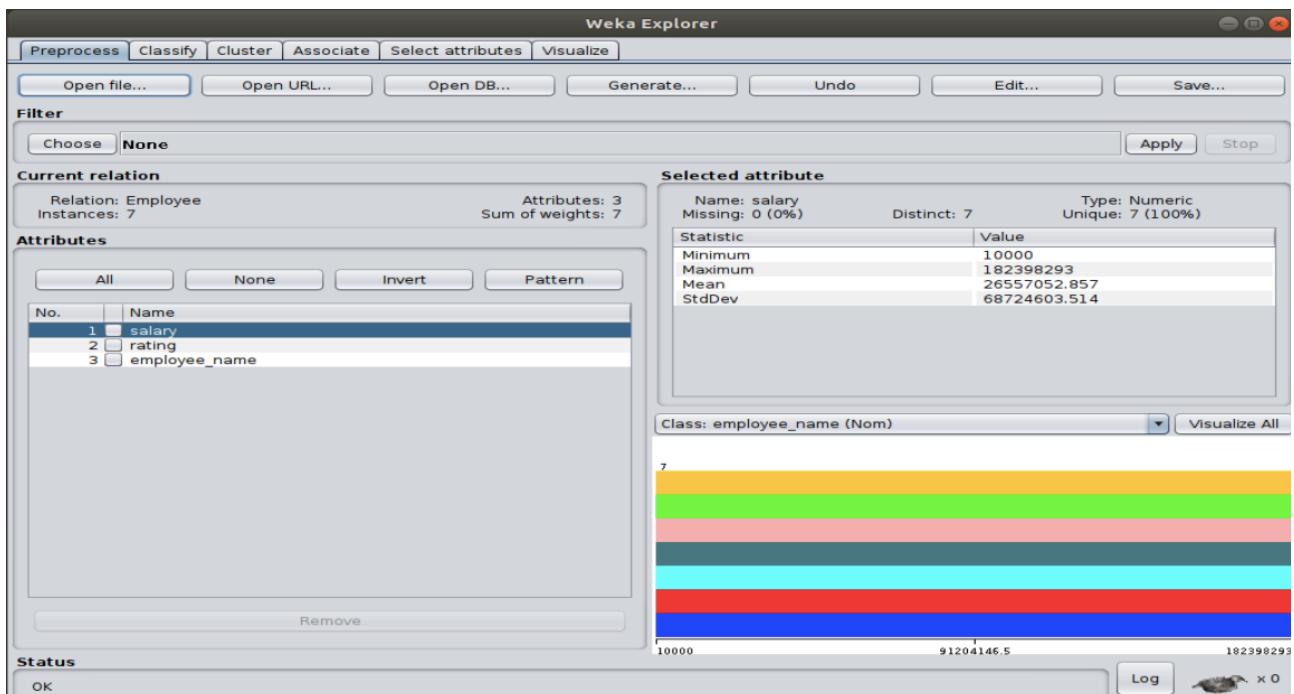


Practical 2 : Apply pre-processing to the training data-set of Employee table.

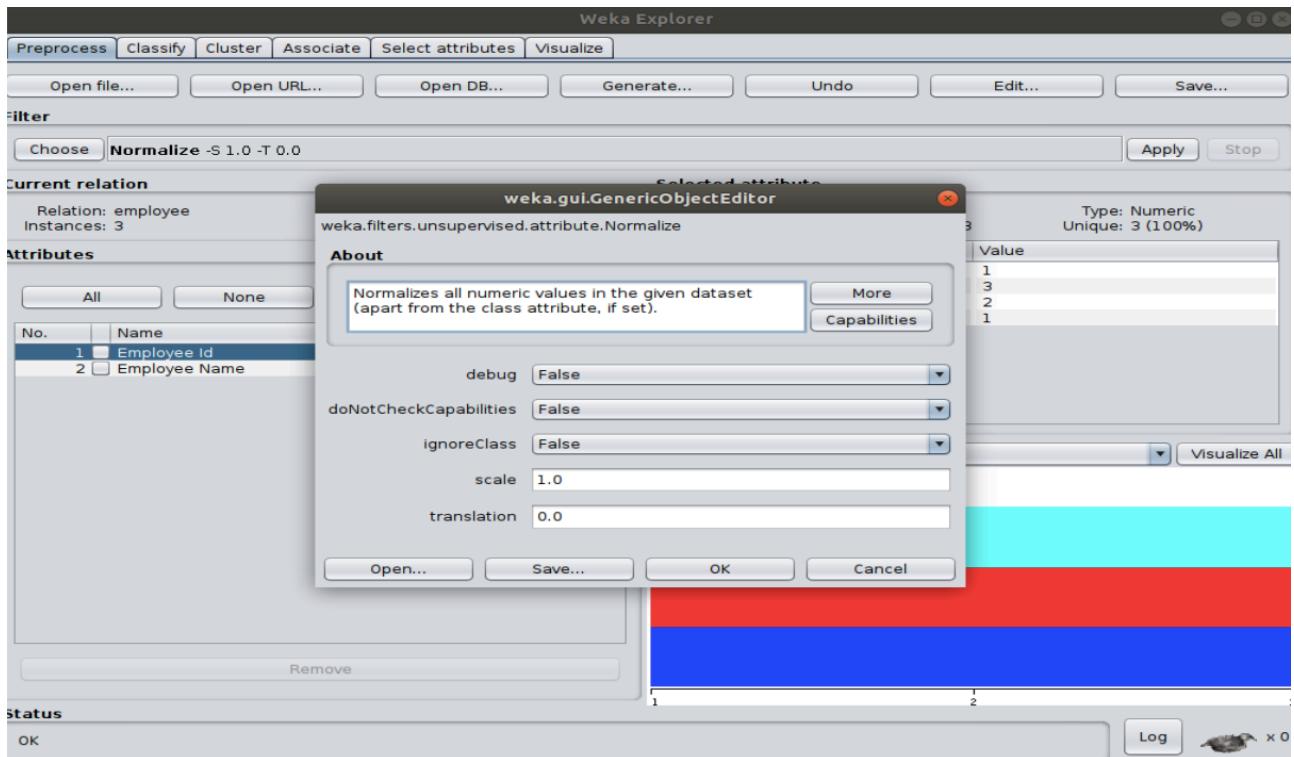
1. Open Weka Explorer.



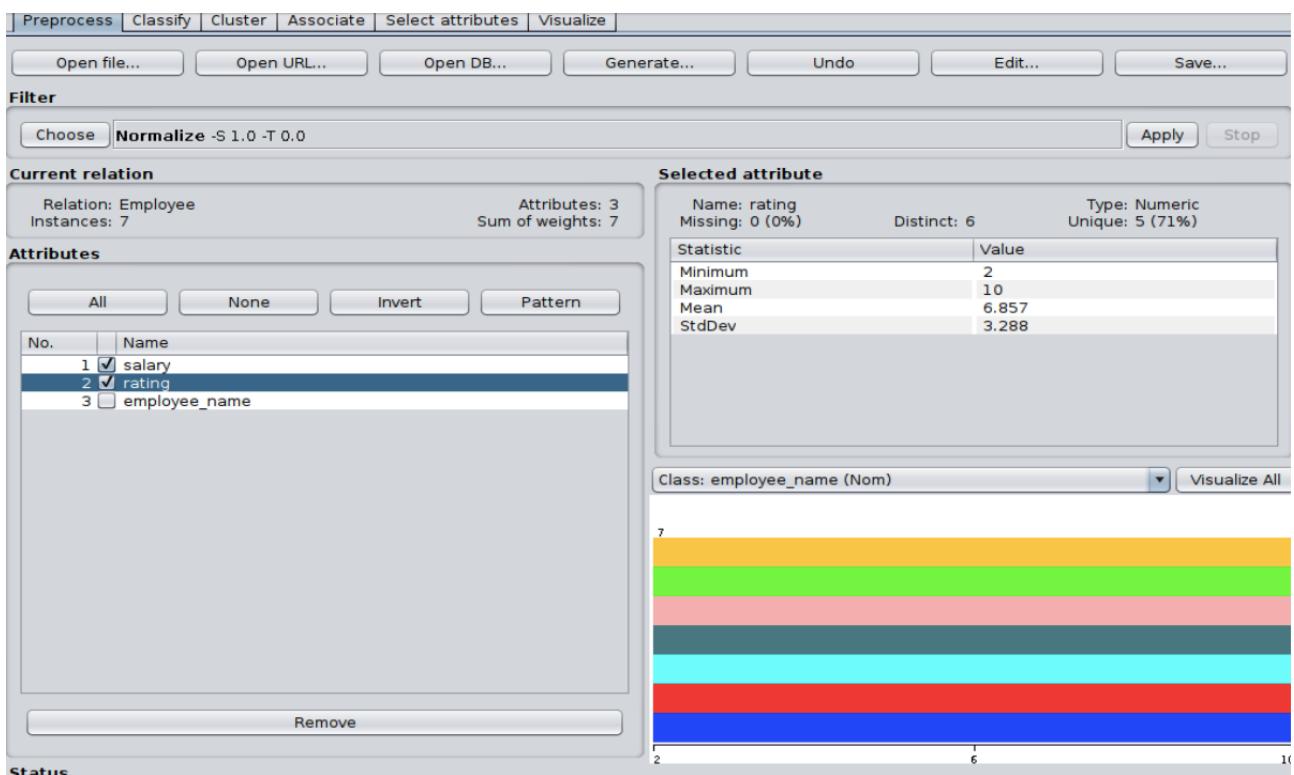
2. Open employee.arff table.



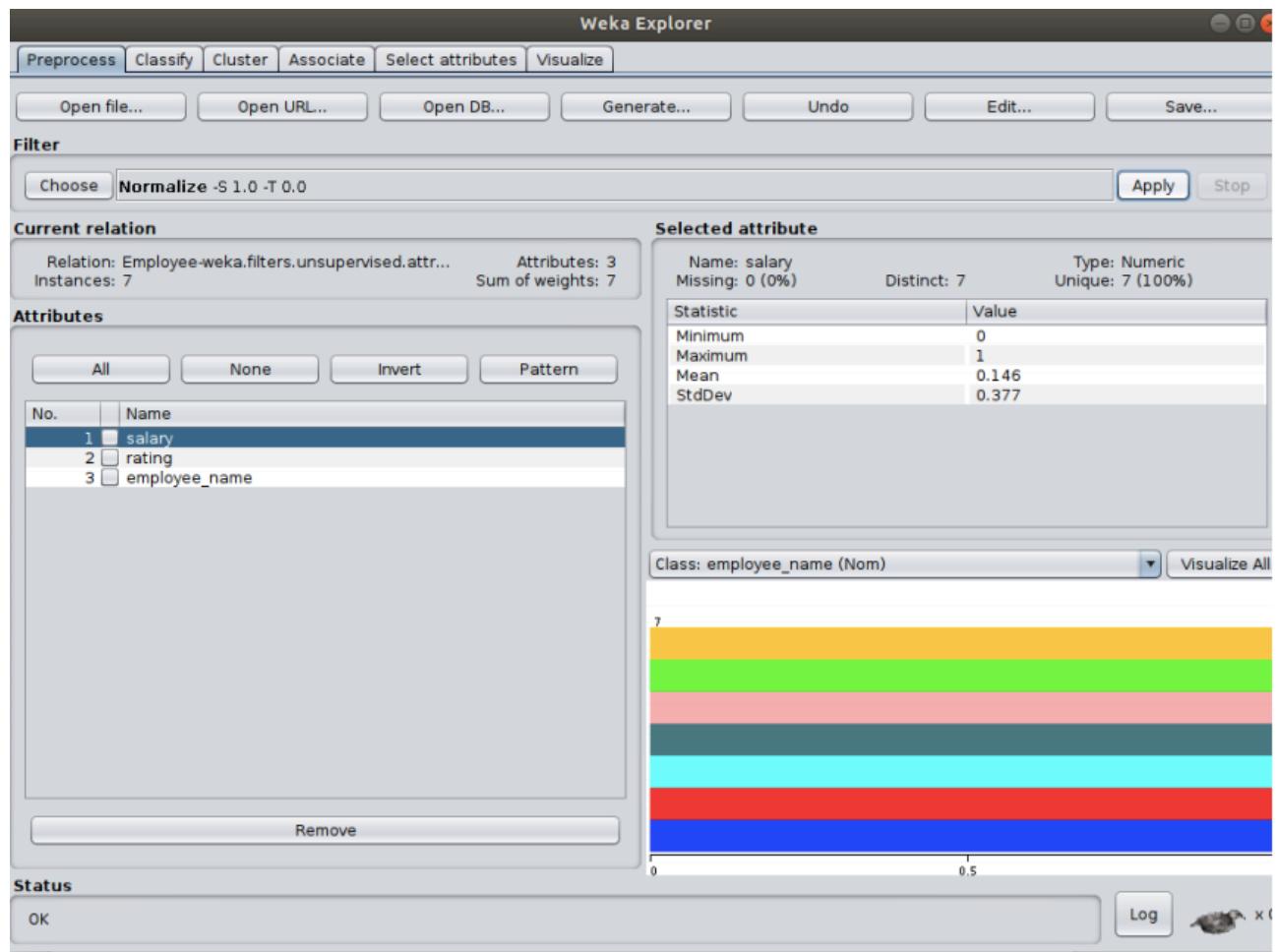
3. Select Normalize from Filter → Unsupervised Learning → Attributes → Normalize



4. Select the attributes you want to Normalize.



5. Click on apply to normalize the selected attributes

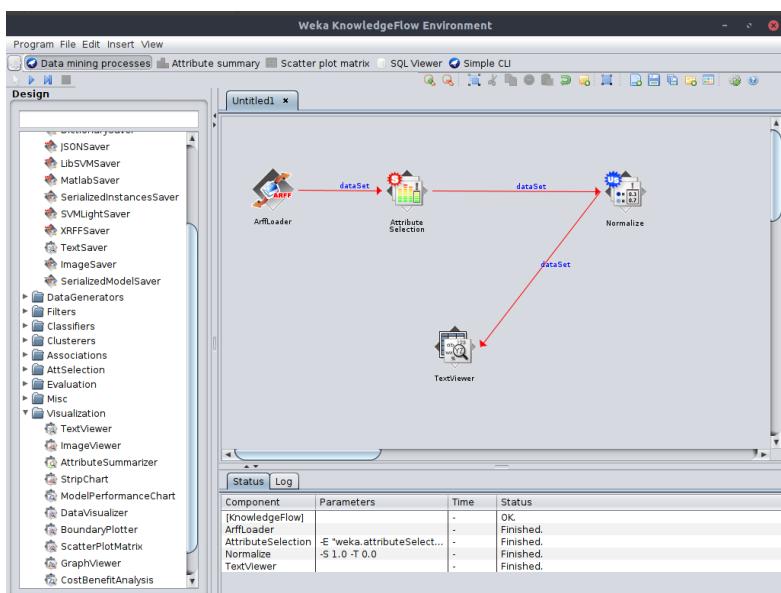


Practical 3 : Normalize Banking Table data using knowledge flow

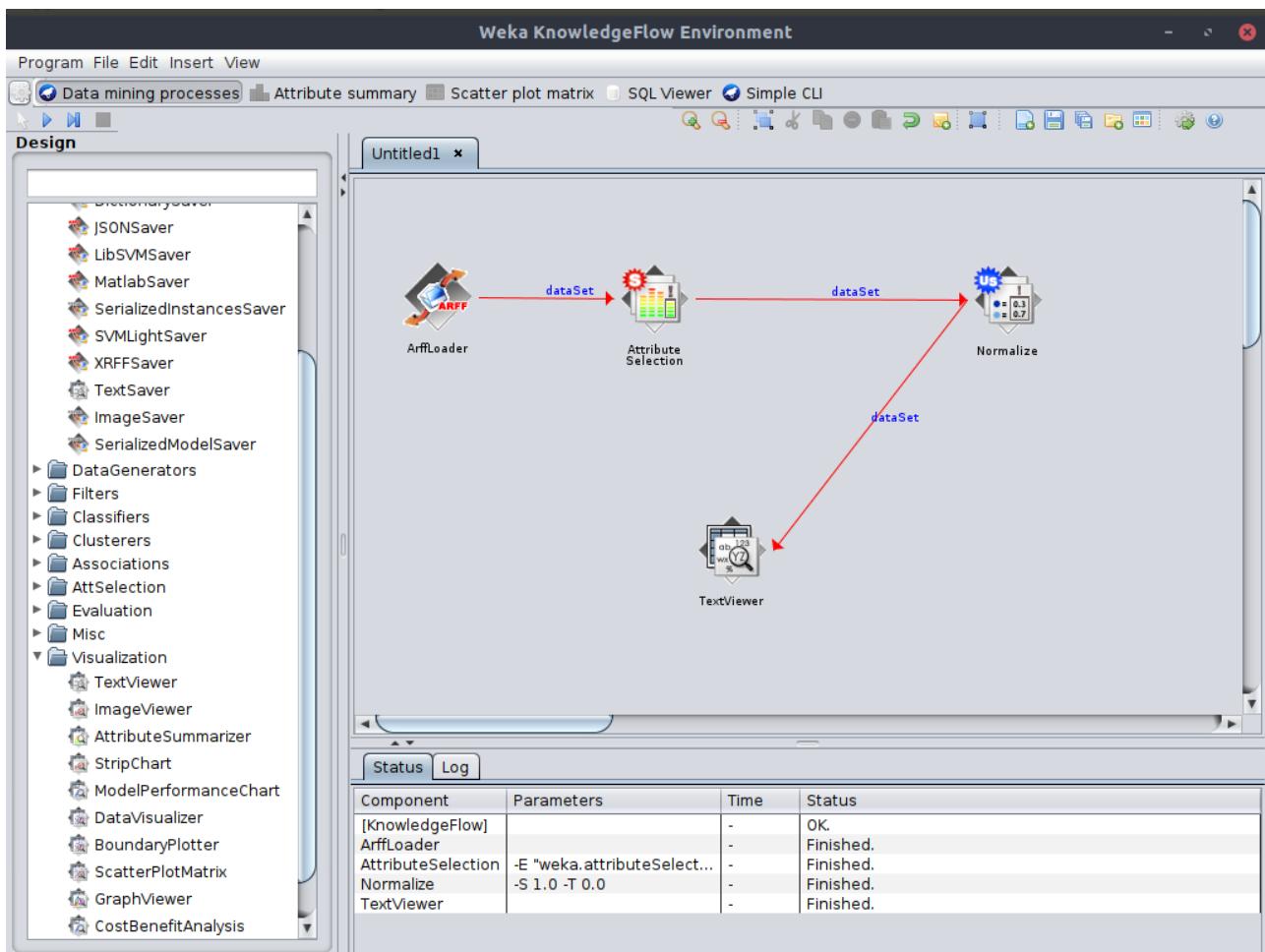
1. Open the weka mining tool and go to knowledge-flow tab



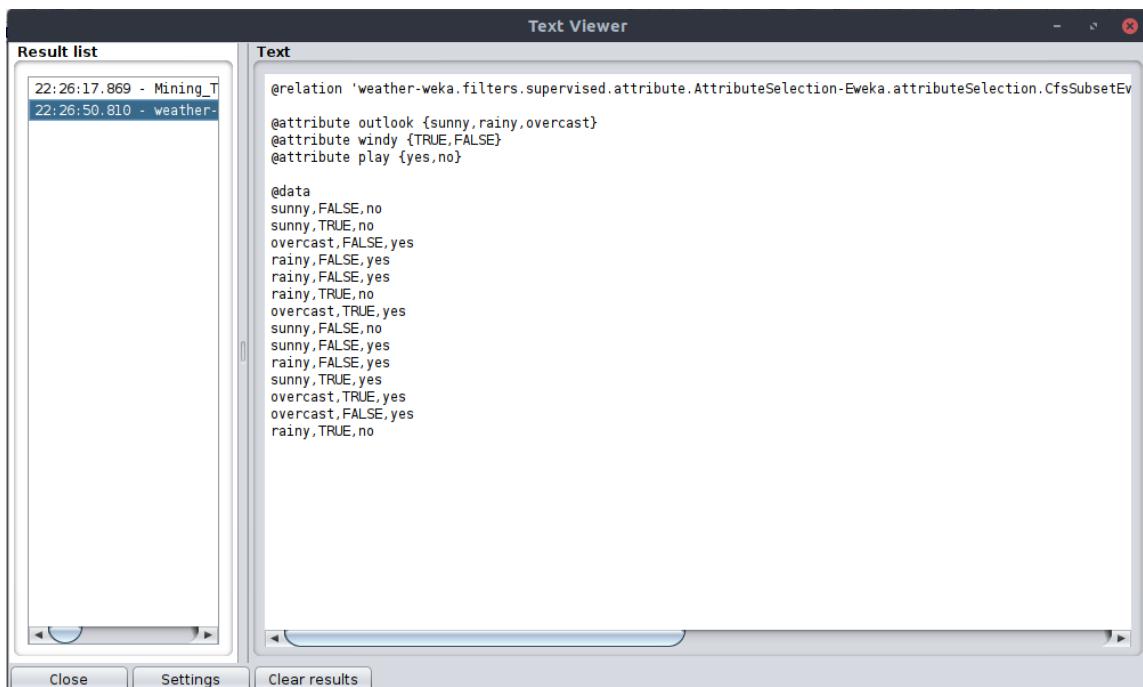
2. Select **ArffLoader** from **DataSources** and place it.
3. Right-click on ArffLoader and configure the **.arff** file in it
4. Select **AttributeSelection** from **Filters -> supervised -> attribute**
5. Select **Normalize** from **Filters -> unsupervised -> attribute**
6. Select **TextViewer** from **Visualization**



7. Run Flow by clicking play button in top left



8. Select **ShowResults** by right-clicking on **TextViewer**. Output is the **Normalized .arff data**.



Output:

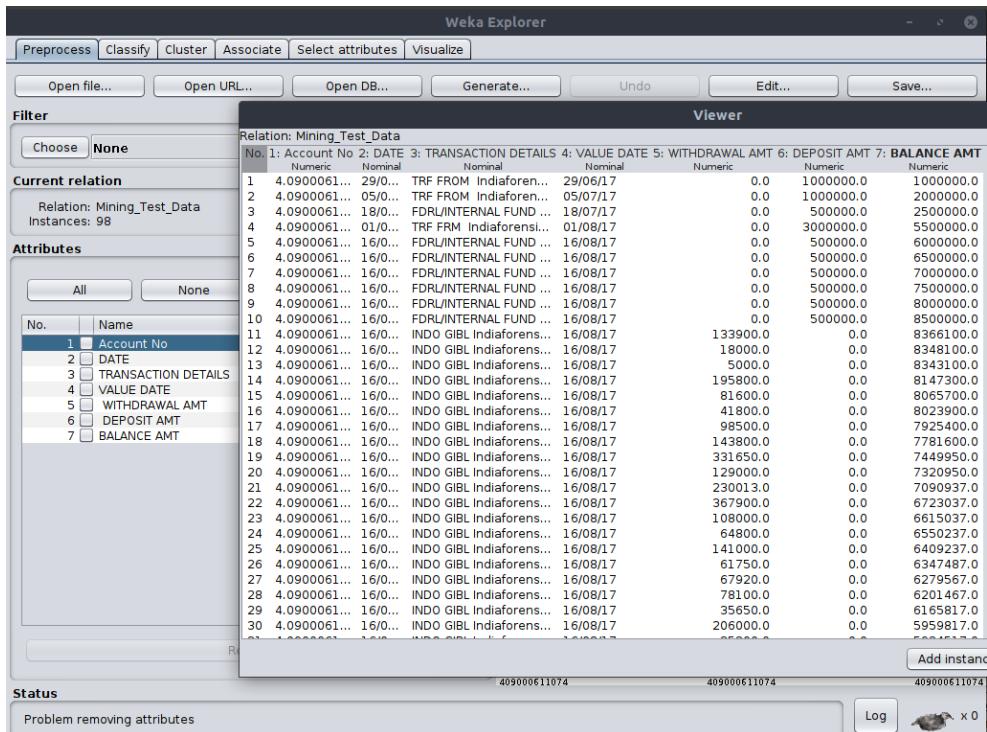
```
@relation 'weather-weka.filters.supervised.attribute.AttributeSelection-
Eweka.attributeSelection.CfsSubsetEval -P 1 -E 1-Sweka.attributeSelection.BestFirst -D
1 -N 5-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0'

@attribute outlook {sunny,rainy,overcast}
@attribute windy {TRUE,FALSE}
@attribute play {yes,no}

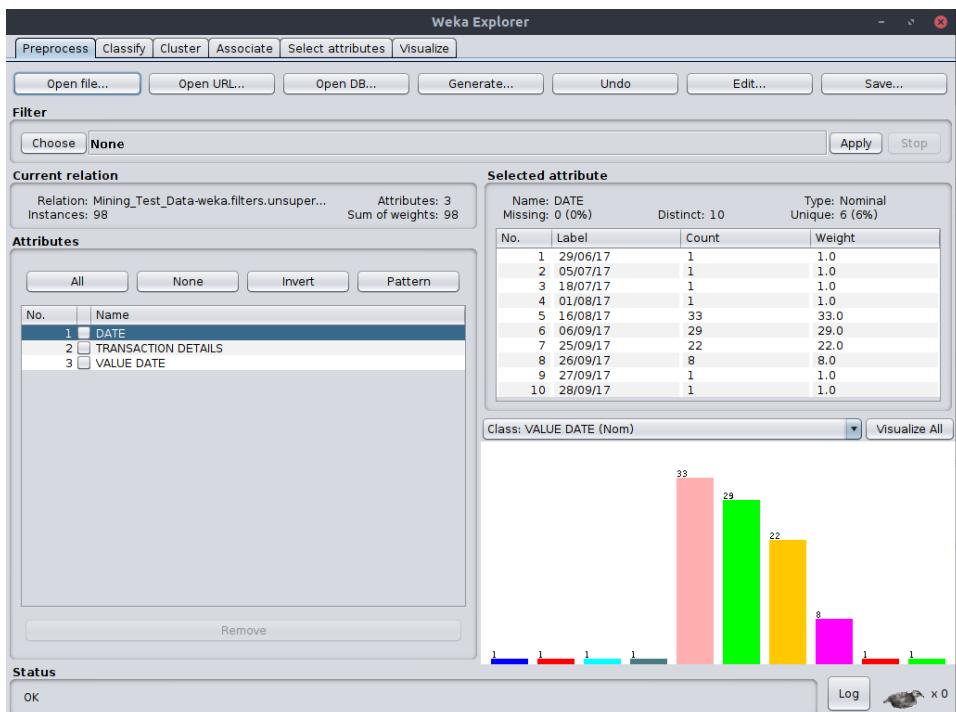
@data
sunny, FALSE, no
sunny, TRUE, no
overcast, FALSE, yes
rainy, FALSE, yes
rainy, FALSE, yes
rainy, TRUE, no
overcast, TRUE, yes
sunny, FALSE, no
sunny, FALSE, yes
rainy, FALSE, yes
sunny, TRUE, yes
overcast, TRUE, yes
overcast, FALSE, yes
rainy, TRUE, no
```

Practical 4 : Find association rules for banking data using weka

1. Open the .arff file in weka mining tool



2. Remove numeric attributes as only nominal attributes are used in apriori association rule mining



3. Click on Associate tab and select apriori algorithm
4. Start association and collect the association rules mined.

==== Run information ===

Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

Relation: Mining_Test_Data-weka.filters.unsupervised.attribute.Remove-R1,5-7

Instances: 98

Attributes: 3

DATE

TRANSACTION DETAILS

VALUE DATE

==== Associator model (full training set) ===

Apriori

=====

Minimum support: 0.1 (10 instances)

Minimum metric <confidence>: 0.9

Number of cycles performed: 18

Generated sets of large itemsets:

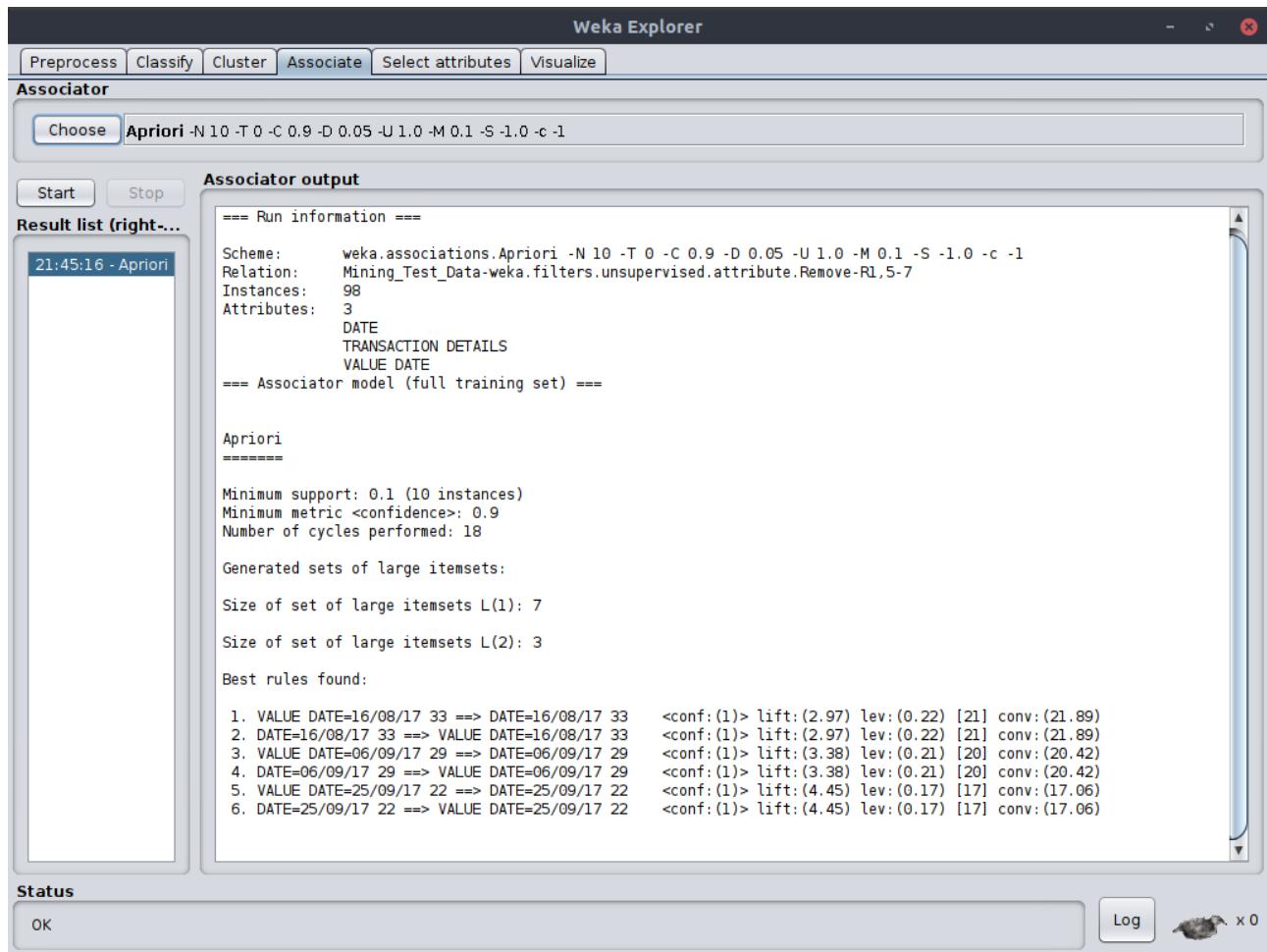
Size of set of large itemsets L(1): 7

Size of set of large itemsets L(2): 3

Best rules found:

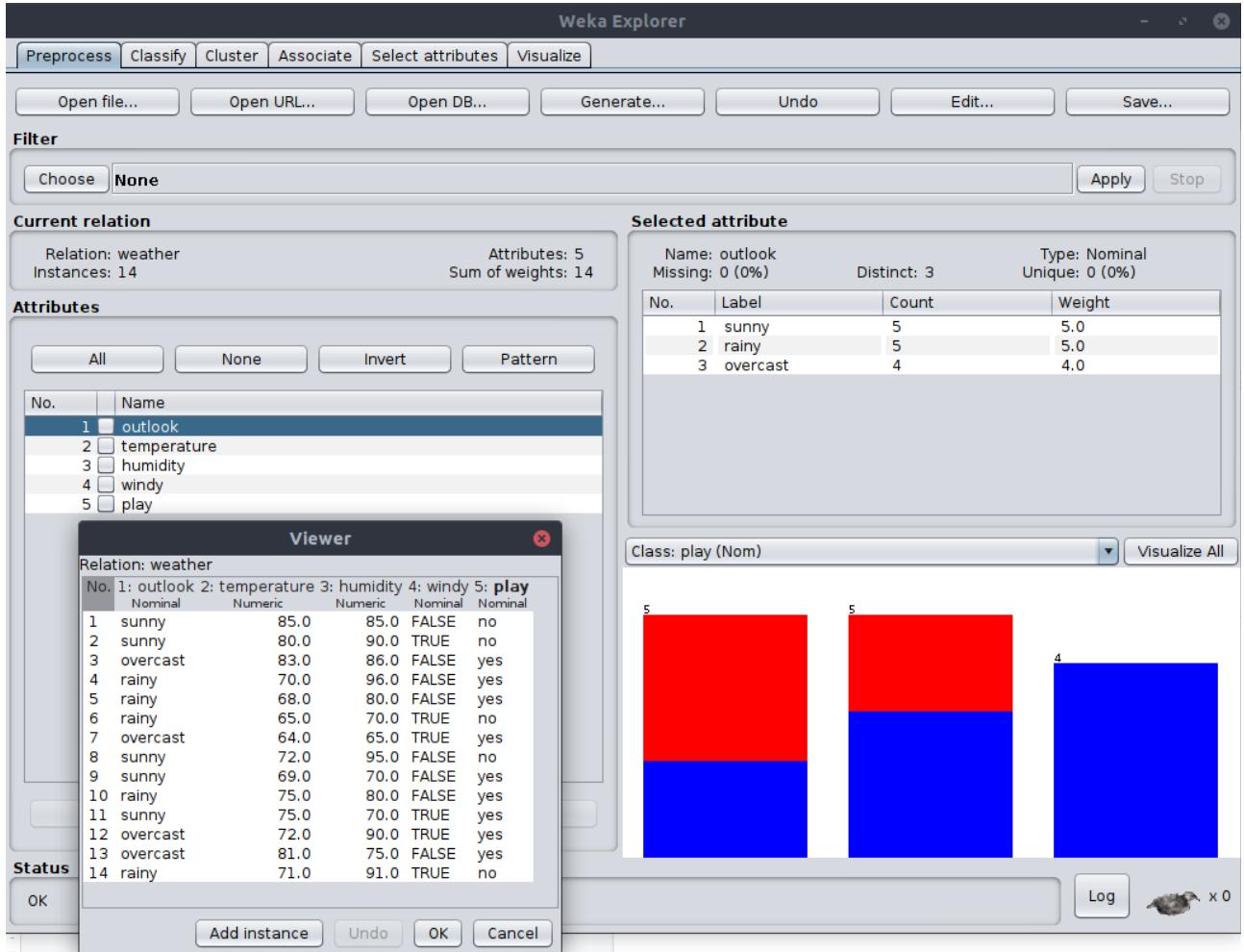
1. VALUE DATE=16/08/17 33 ==> DATE=16/08/17 33 <conf:(1)> lift:(2.97) lev:(0.22) [21] conv:(21.89)
2. DATE=16/08/17 33 ==> VALUE DATE=16/08/17 33 <conf:(1)> lift:(2.97) lev:(0.22) [21] conv:(21.89)
3. VALUE DATE=06/09/17 29 ==> DATE=06/09/17 29 <conf:(1)> lift:(3.38) lev:(0.21) [20] conv:(20.42)
4. DATE=06/09/17 29 ==> VALUE DATE=06/09/17 29 <conf:(1)> lift:(3.38) lev:(0.21) [20] conv:(20.42)
5. VALUE DATE=25/09/17 22 ==> DATE=25/09/17 22 <conf:(1)> lift:(4.45) lev:(0.17) [17] conv:(17.06)

6. DATE=25/09/17 22 ==> VALUE DATE=25/09/17 22 <conf:(1)> lift:(4.45) lev:(0.17) [17] conv:(17.06)



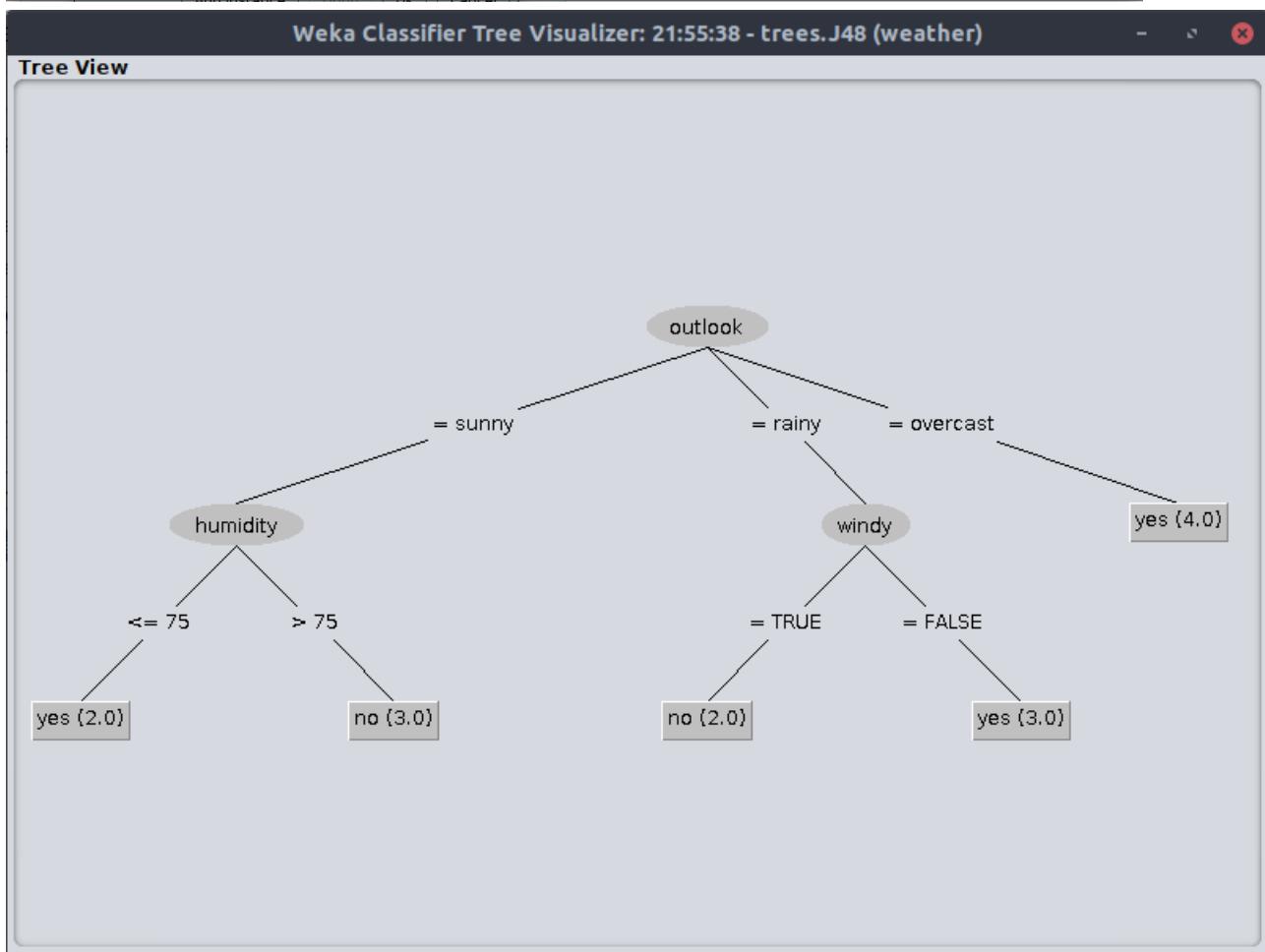
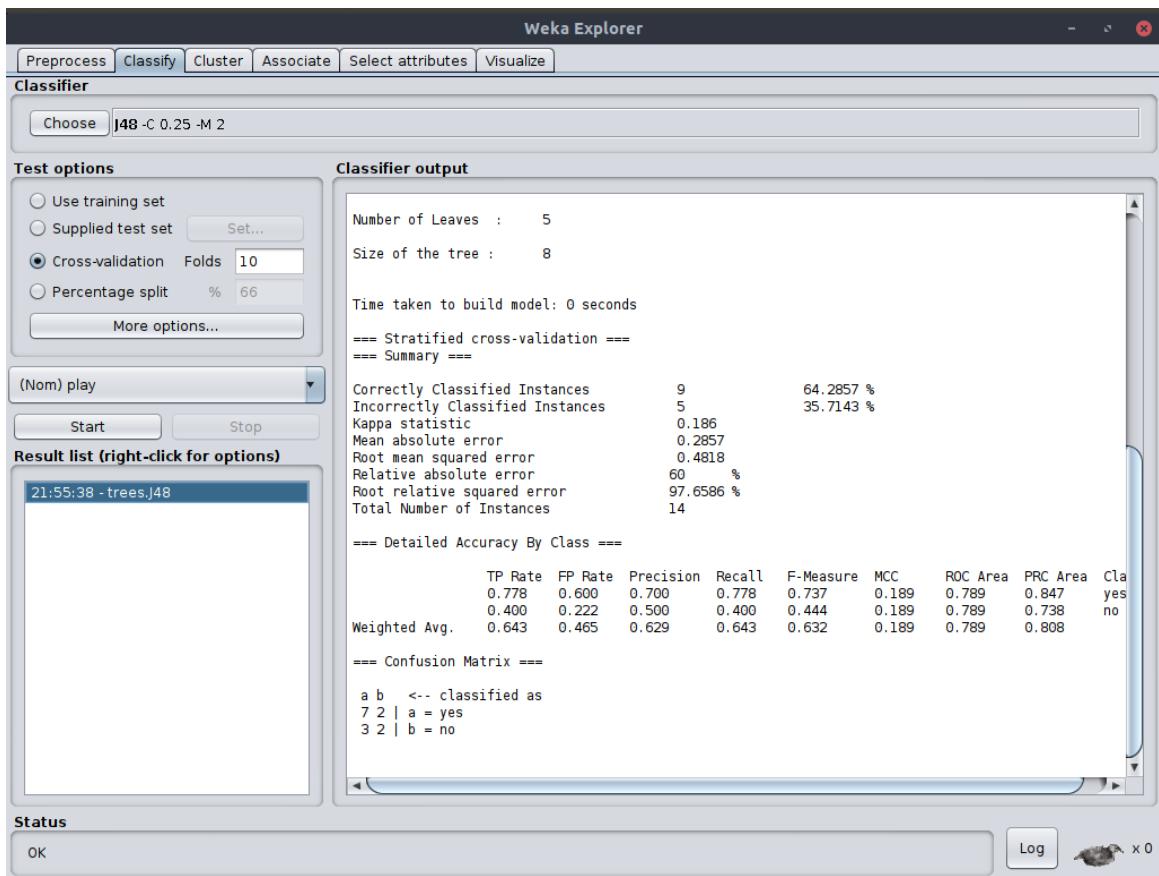
Practical 5 : To construct Decision tree for Weather data in weka

Step-1) Open the .arff file in weka mining tool



Step-2) Click on Classify tab and select J48 from the trees tab of the algorithms available

Step-3) Select Play attribute, Start algorithm



==== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: weather

Instances: 14

Attributes: 5

- outlook
- temperature
- humidity
- windy
- play

Test mode: 10-fold cross-validation

==== Classifier model (full training set) ===

J48 pruned tree

- outlook = sunny
 - | humidity <= 75: yes (2.0)
 - | humidity > 75: no (3.0)
- outlook = rainy
 - | windy = TRUE: no (2.0)
 - | windy = FALSE: yes (3.0)
- outlook = overcast: yes (4.0)

Number of Leaves : 5

Size of the tree : 8

Time taken to build model: 0 seconds

==== Stratified cross-validation ===

==== Summary ===

Correctly Classified Instances	9	64.2857 %
Incorrectly Classified Instances	5	35.7143 %
Kappa statistic	0.186	
Mean absolute error	0.2857	
Root mean squared error	0.4818	
Relative absolute error	60	%

Root relative squared error 97.6586 %

Total Number of Instances 14

==== Detailed Accuracy By Class ===

Area	Class	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC
	yes	0.778	0.600	0.700	0.778	0.737	0.189	0.789	0.847
	no	0.400	0.222	0.500	0.400	0.444	0.189	0.789	0.738
Weighted Avg.		0.643	0.465	0.629	0.643	0.632	0.189	0.789	0.808

==== Confusion Matrix ===

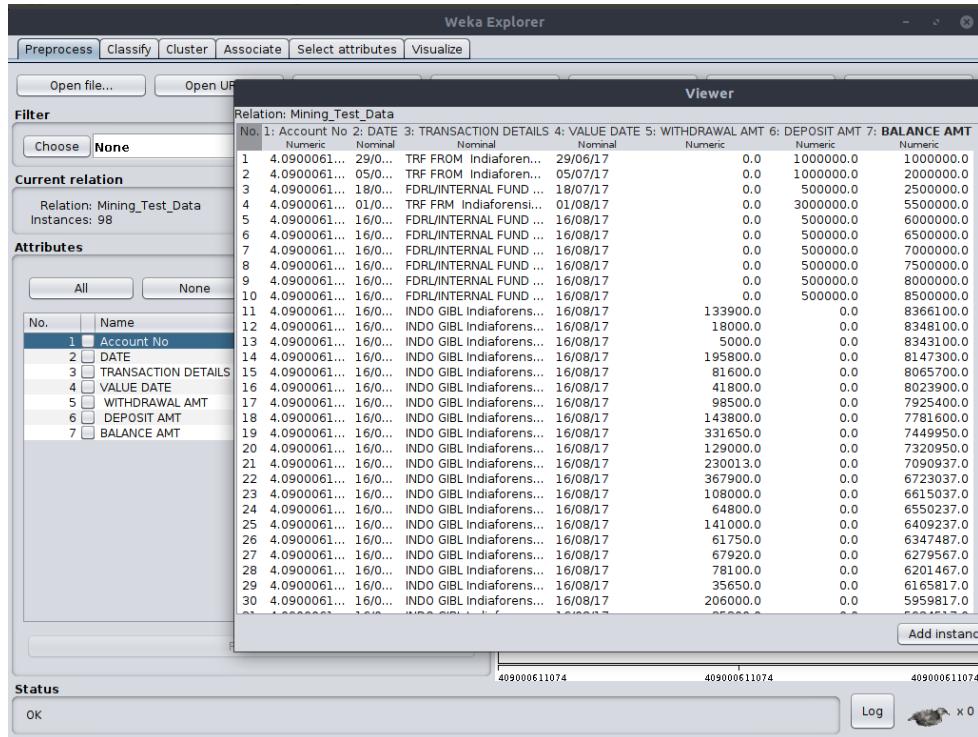
a b <-- classified as

7 2 | a = yes

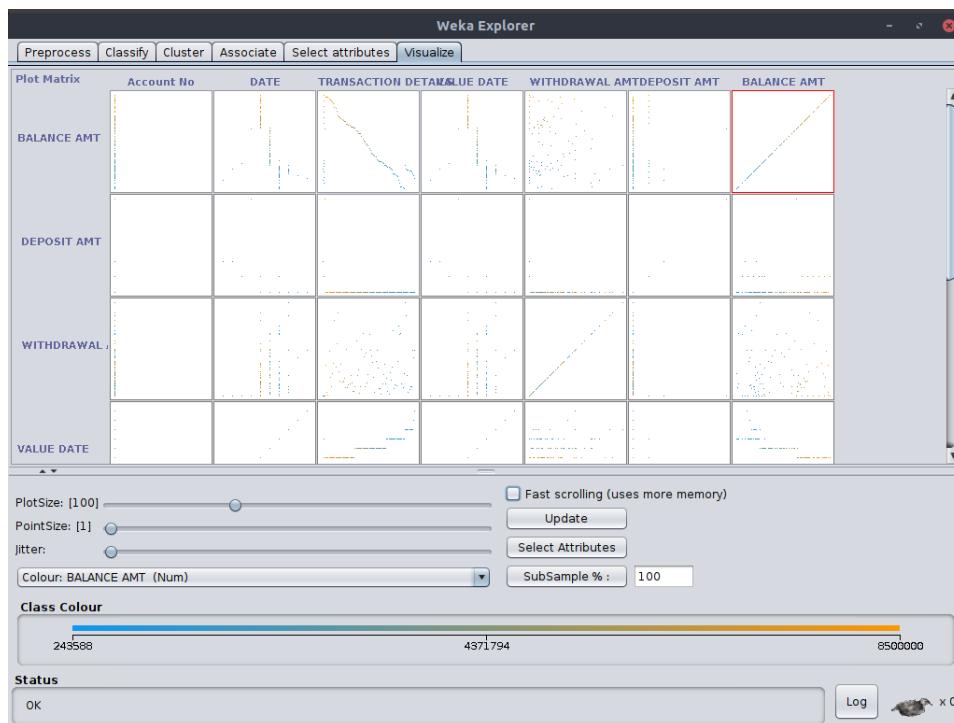
3 2 | b = no

Practical 6 : Write a procedure for visualization of weather table

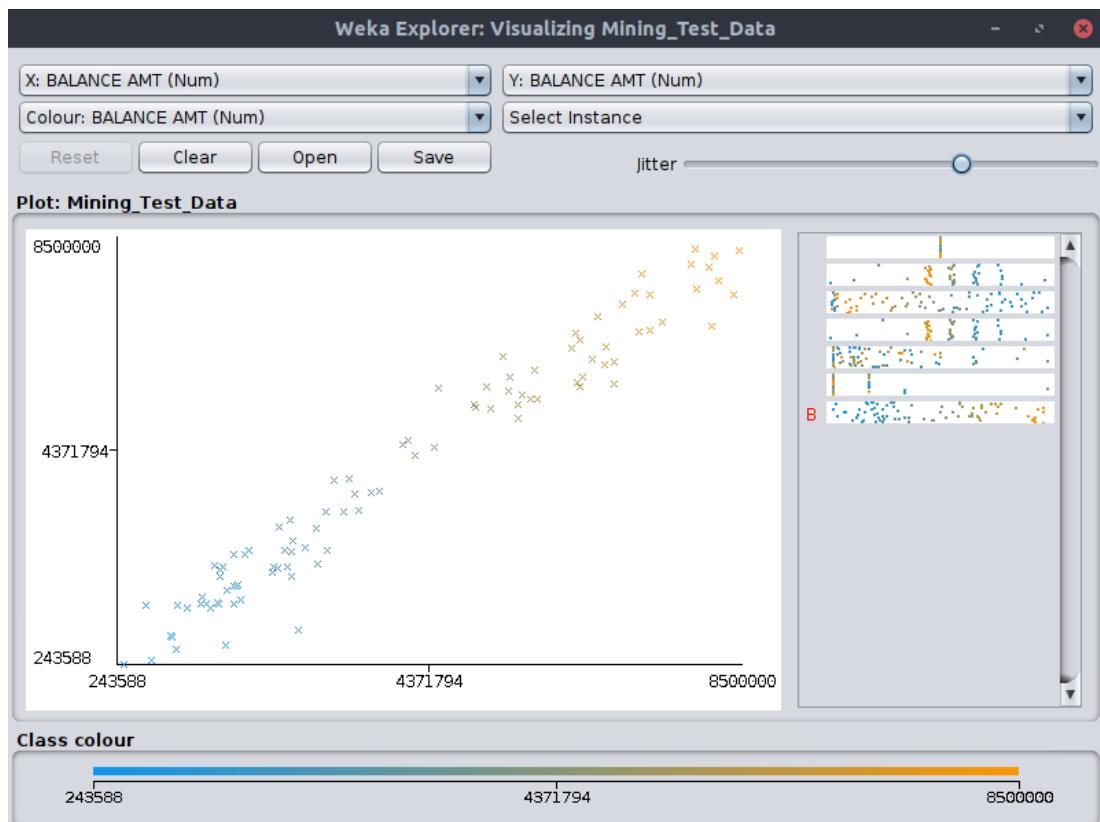
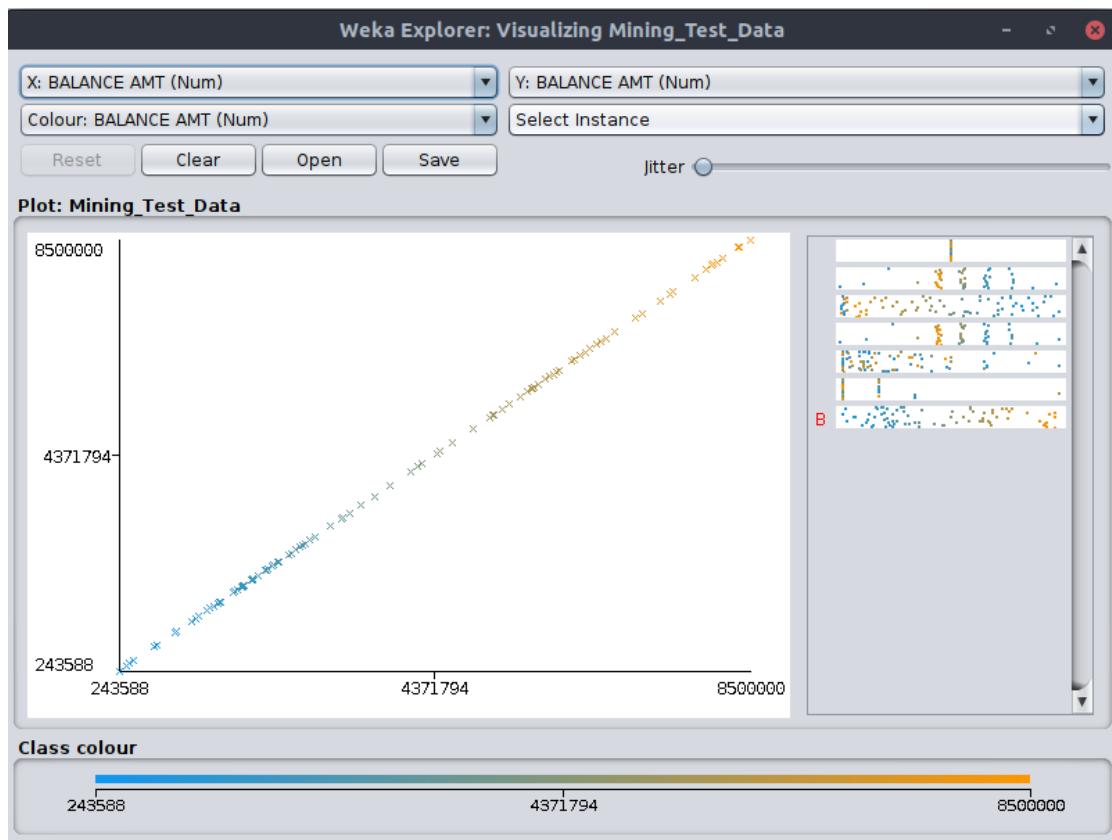
1. Open the .arff file in weka mining tool

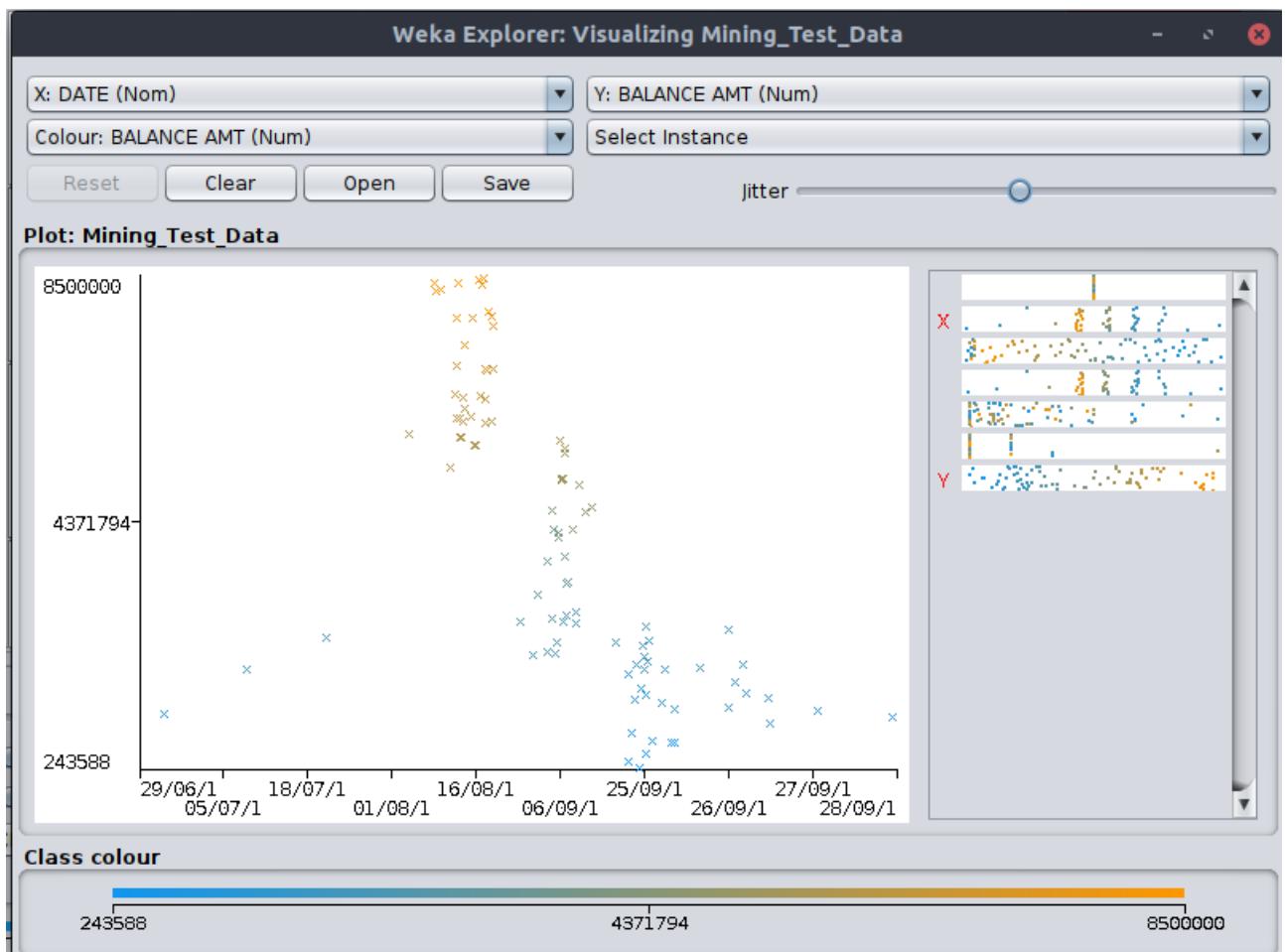


2. Click on the visualize tab and see various attributes and their visualization

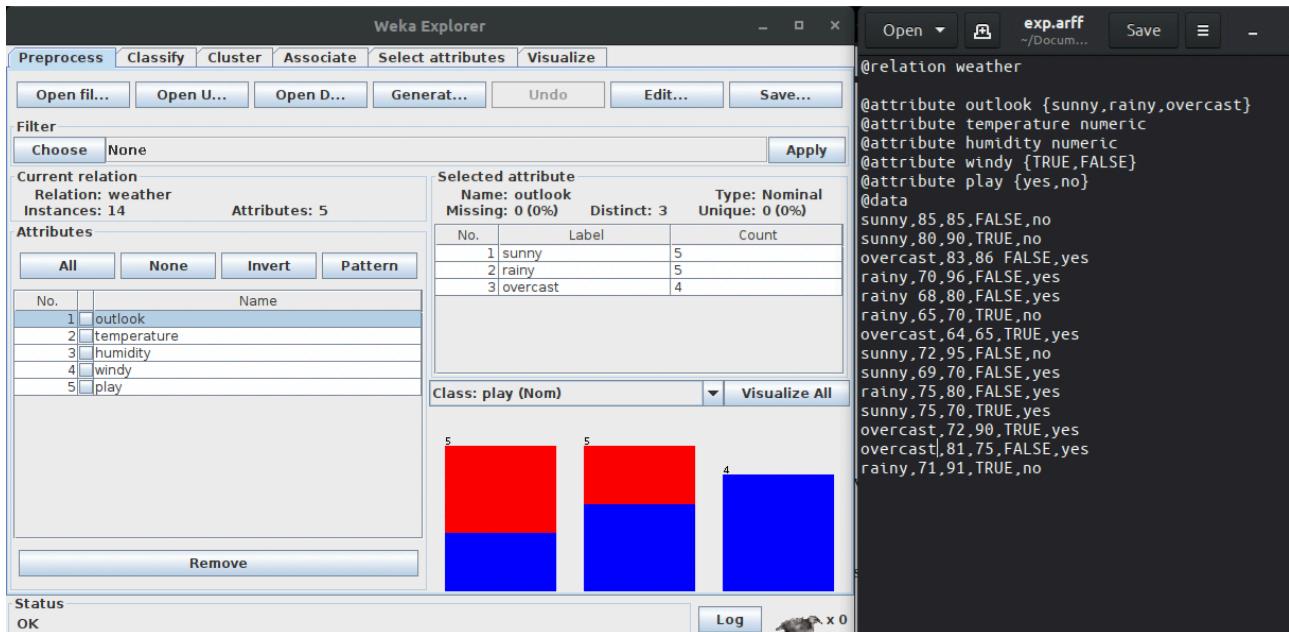


3. Select an attribute and visualize it with various jitter levels.





Practical 7 : Write a procedure for cross validation using J48 algo for weather table



Text Viewer

Result list

```

21:26:21 - J48
==== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: weather

Correctly Classified Instances      9          64.2857 %
Incorrectly Classified Instances   5          35.7143 %
Kappa statistic                   0.186
Mean absolute error               0.2857
Root mean squared error           0.4818
Relative absolute error           60          %
Root relative squared error      97.6586 %
Total Number of Instances         14

==== Detailed Accuracy By Class ===

          TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area   Class
          0.778     0.6       0.7       0.778     0.737     0.789     yes
          0.4       0.222    0.5       0.4       0.444     0.789     no
Weighted Avg.      0.643     0.465    0.629     0.643     0.632     0.789

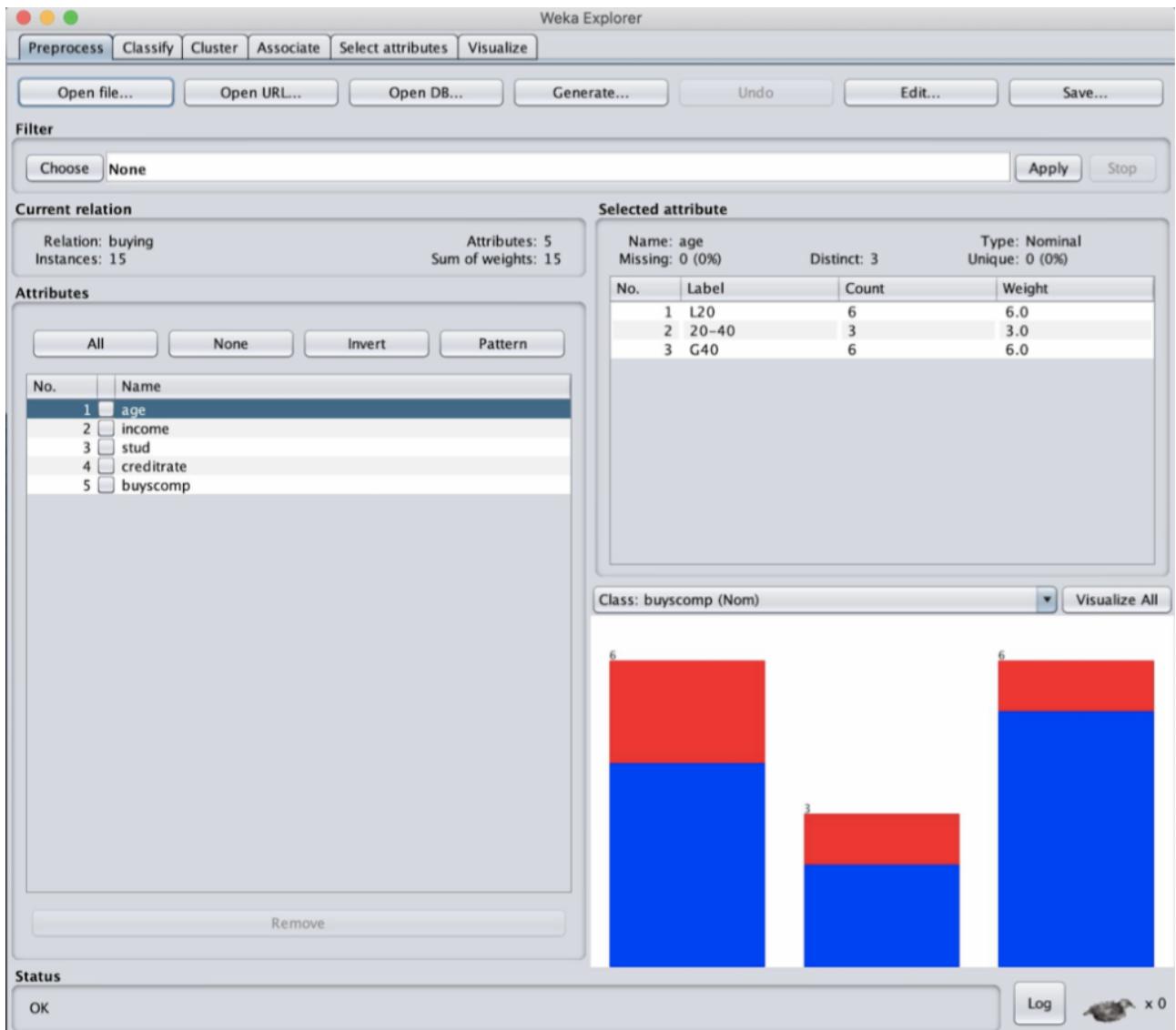
==== Confusion Matrix ===

a b  <-- classified as
7 2 | a = yes
3 2 | b = no

```

Practical 8 : Write a procedure for Clustering Buying data using Cobweb Algorithm.

1. Data Loading



2. Review of Loaded Data

Viewer

Relation: buying

No.	1: age	2: income	3: stud	4: creditrate	5: buyscomp
	Nominal	Nominal	Nominal	Nominal	Nominal
1	L20	high	no	fair	yes
2	20-...	low	yes	fair	yes
3	G40	medium	yes	fair	yes
4	L20	low	no	fair	no
5	G40	high	no	excellent	yes
6	L20	low	yes	fair	yes
7	20-...	high	yes	excellent	no
8	G40	low	no	fair	yes
9	L20	high	yes	excellent	yes
...	G40	high	no	fair	yes
...	L20	low	yes	excellent	no
...	G40	high	yes	excellent	no
...	20-...	medium	yes	excellent	yes
...	L20	medium	yes	fair	yes
...	G40	high	yes	excellent	yes

Add instance Undo OK Cancel

3. Result after COBWEB algorithm

Weka Explorer

Clusterer

Choose Cobweb -A 1.0 -C 0.0028209479177387815 -S 42

Cluster mode

- Use training set
- Supplied test set Set...
- Percentage split % 66
- Classes to clusters evaluation (Nom) buyscomp
- Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

17:33:21 – Cobweb

Clusterer output

```
== Run information ==
Scheme: weka.clusterers.Cobweb -A 1.0 -C 0.0028209479177387815 -S 42
Relation: buying
Instances: 15
Attributes: 5
age
income
stud
creditrate
buyscomp
Test mode: evaluate on training data

== Clustering model (full training set) ==
Number of merges: 0
Number of splits: 0
Number of clusters: 24

node 0 [15]
| node 1 [5]
| | leaf 2 [1]
| node 1 [5]
| | node 3 [2]
| | | leaf 4 [1]
| | node 3 [2]
| | | leaf 5 [1]
| node 1 [5]
| | node 6 [2]
| | | leaf 7 [1]
| | node 6 [2]
| | | leaf 8 [1]
node 0 [15]
| node 9 [5]
| | leaf 10 [1]
| node 9 [5]
| | node 11 [2]
| | | leaf 12 [1]
| | node 11 [2]
| | | leaf 13 [1]
node 9 [5]
| node 14 [2]
```

Weka Explorer

Clusterer

Choose Cobweb -A 1.0 -C 0.0028209479177387815 -S 42

Cluster mode

- Use training set
- Supplied test set Set...
- Percentage split % 66
- Classes to clusters evaluation (Nom) buyscomp
- Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

17:33:21 – Cobweb

Clusterer output

```
| node 11 [2]
| | leaf 13 [1]
node 9 [5]
| node 14 [2]
| | leaf 15 [1]
| | node 14 [2]
| | | leaf 16 [1]
node 0 [15]
| node 17 [5]
| | node 18 [2]
| | | leaf 19 [1]
| | node 18 [2]
| | | leaf 20 [1]
| node 17 [5]
| | leaf 21 [1]
node 17 [5]
| | leaf 22 [1]
node 17 [5]
| | leaf 23 [1]
```

Time taken to build model (full training data) : 0.01 seconds

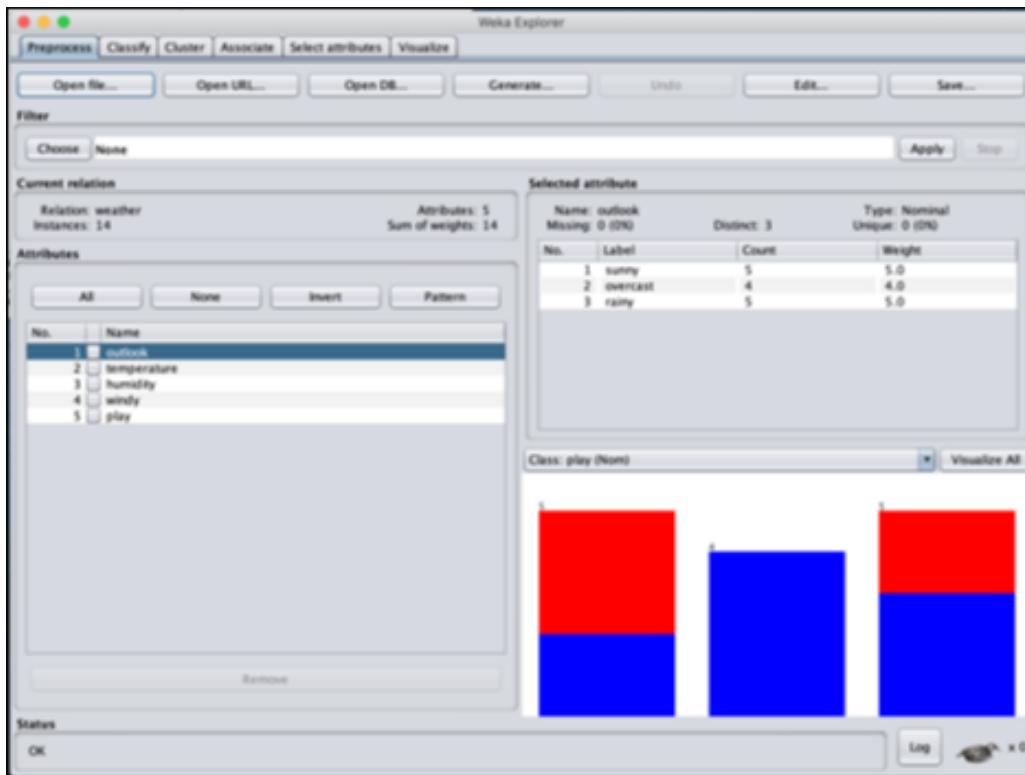
== Model and evaluation on training set ==

Clustered Instances

2	1 (7%)
4	1 (7%)
5	1 (7%)
7	1 (7%)
8	1 (7%)
10	1 (7%)
12	1 (7%)
13	1 (7%)
15	1 (7%)
16	1 (7%)
19	1 (7%)
20	1 (7%)
21	1 (7%)
22	1 (7%)
23	1 (7%)

Practical 9 : Write a procedure for Clustering Weather data using EM Algorithm.

1. Data Loading



2. Review of Loaded Data

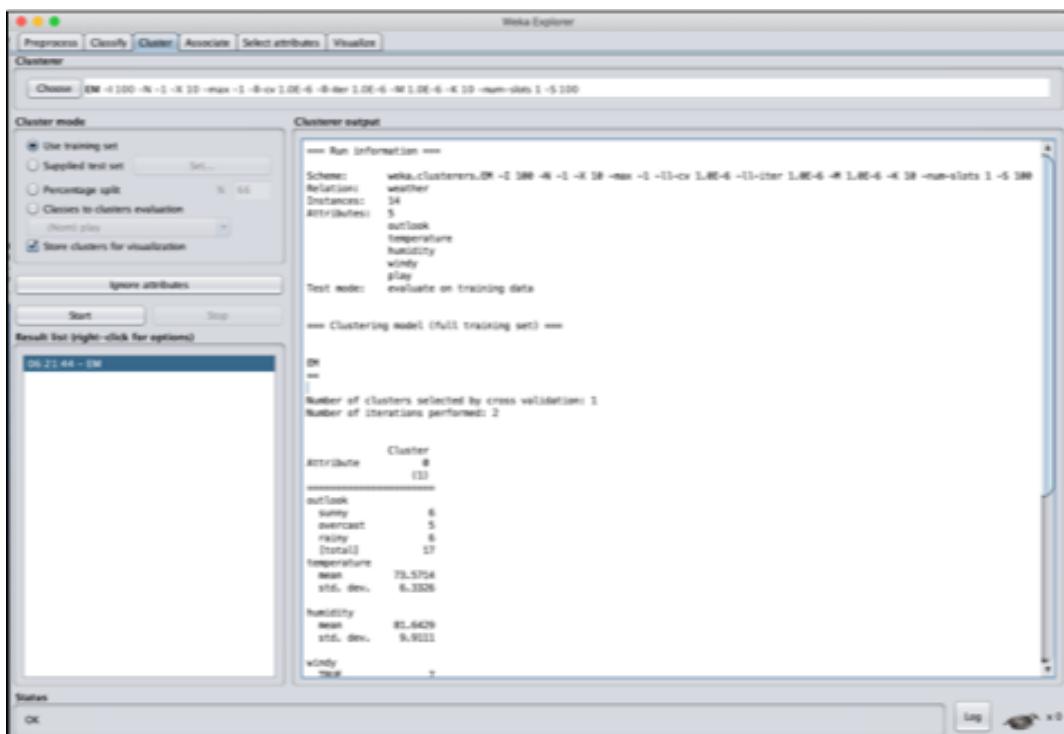
Viewer

Relation: weather

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	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
...	rainy	75.0	80.0	FALSE	yes
...	sunny	75.0	70.0	TRUE	yes
...	overcast	72.0	90.0	TRUE	yes
...	overcast	81.0	75.0	FALSE	yes
...	rainy	71.0	91.0	TRUE	no

Add instance Undo OK Cancel

3. Result after EM algorithm



Clusterer output

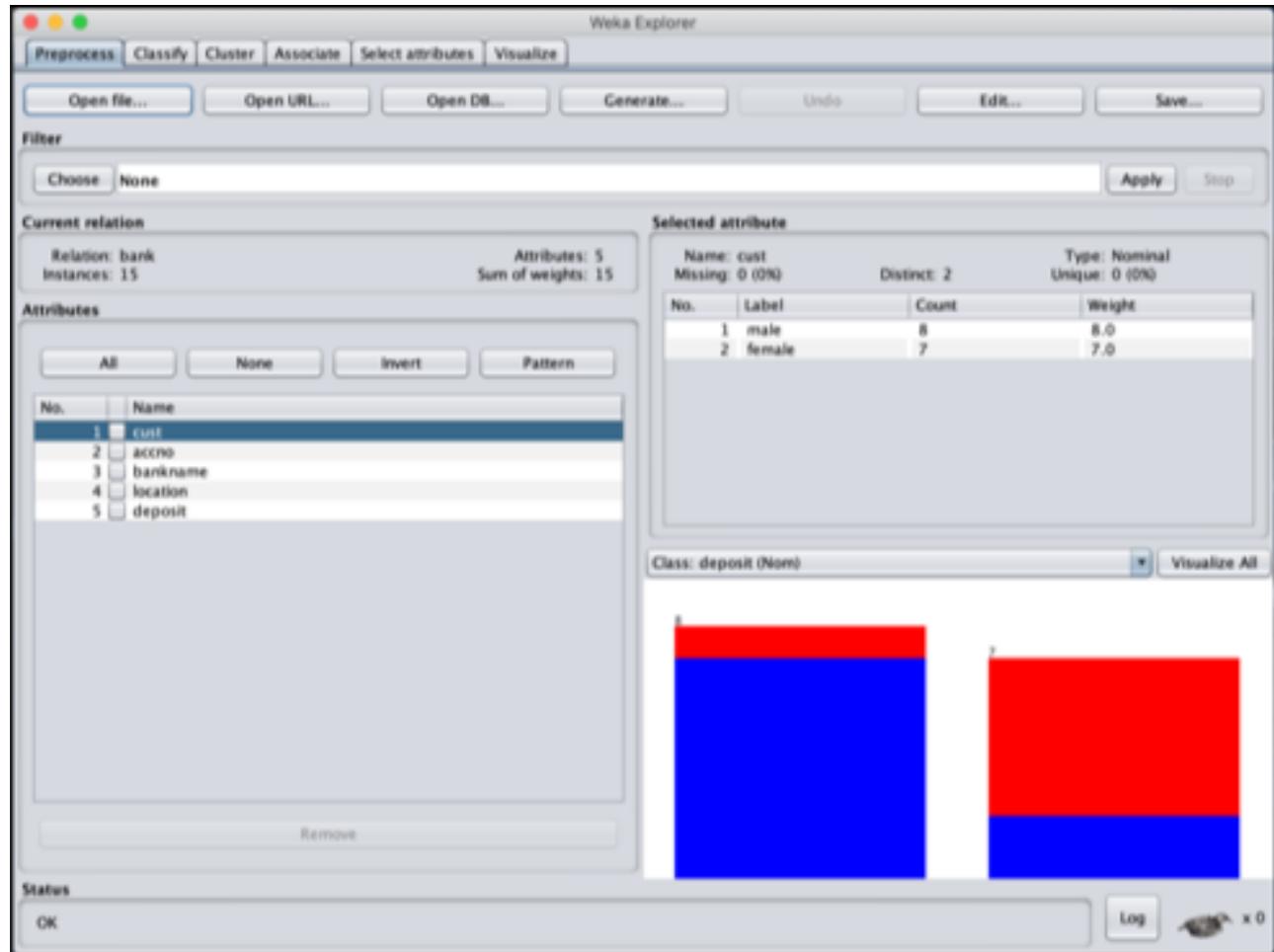
```
NUMBER OF CLUSTERS SELECTED BY CROSS VALIDATION: 1
Number of iterations performed: 2

Cluster
Attribute      0
(outlook)
=====
outlook
  sunny          6
  overcast       5
  rainy          6
  [total]        17
temperature
  mean         73.5714
  std. dev.   6.3326
humidity
  mean         81.6429
  std. dev.   9.9111
windy
  TRUE         7
  FALSE        9
  [total]      16
play
  yes          10
  no           6
  [total]      16

Time taken to build model (full training data) : 0.06 seconds
==== Model and evaluation on training set ====
Clustered Instances
0      14 (100%)
Log likelihood: -9.4063
```

Practical 10 : Write a procedure for Banking data using Farthest First Algorithm.

1. Data Loading



2. Review of Loaded Data

Relation: bank

No. 1: cust	Nominal
1	male
2	fem...
3	male
4	male
5	fem...
6	male
7	fem...
8	fem...
9	male
...	male
...	fem...
...	male
...	fem...
...	male
...	fem...

```

Final cluster centroids:
Cluster#
Attribute   Full Data      0          1
              (13.0)    (9.0)    (4.0)
=====
eid           107  107.7778  105.25
ename         raj   anil     raj
salary       12769.2308 13222.2222 11750
exp            4.1538   3.8889   4.75
address       kdp      kdp     pdtr

Fitted estimators (with ML estimates of variance):
Cluster: 0 Prior probability: 0.66667
Attribute: eid
Normal Distribution. Mean = 107.7778 StdDev = 3.4247
Attribute: ename
Discrete Estimator. Counts = 1 1 2 2 2 2 2 1 2 1 2 2 2 (Total = 22)
Attribute: salary
Normal Distribution. Mean = 13222.2222 StdDev = 1396.645
Attribute: exp
Normal Distribution. Mean = 3.8889 StdDev = 1.0999
Attribute: address
Discrete Estimator. Counts = 1 7 3 2 (Total = 13)

Cluster: 1 Prior probability: 0.33333
Attribute: eid
Normal Distribution. Mean = 105.25 StdDev = 3.8324
Attribute: ename
Discrete Estimator. Counts = 2 2 1 1 1 1 1 2 1 2 1 1 1 (Total = 17)
Attribute: salary
Normal Distribution. Mean = 11750 StdDev = 1920.2864
Attribute: exp
Normal Distribution. Mean = 4.75 StdDev = 0.433
Attribute: address
Discrete Estimator. Counts = 3 1 1 3 (Total = 8)

```

3. Result after

Time taken to build model (full training data) : 0.01 seconds

== Model and evaluation on training set ==

Clustered Instances

0	9 (69%)
1	4 (31%)

Log likelihood: -16.52967

FarthestFirst

Cluster centroids:

Cluster	Attribute	Value
0	gender	male
0	location	sbi
0	deposit	hyd
0	exp	yes
1	gender	female
1	location	hdfc
1	deposit	jmd
1	exp	no

Test mode: evaluate on training data

== Clustering model (full training set) ==

FarthestFirst

Cluster centroids:

Cluster	Attribute	Value
0	gender	male
0	location	sbi
0	deposit	hyd
0	exp	yes
1	gender	female
1	location	hdfc
1	deposit	jmd
1	exp	no

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

== Model and evaluation on training set ==

Clustered Instances

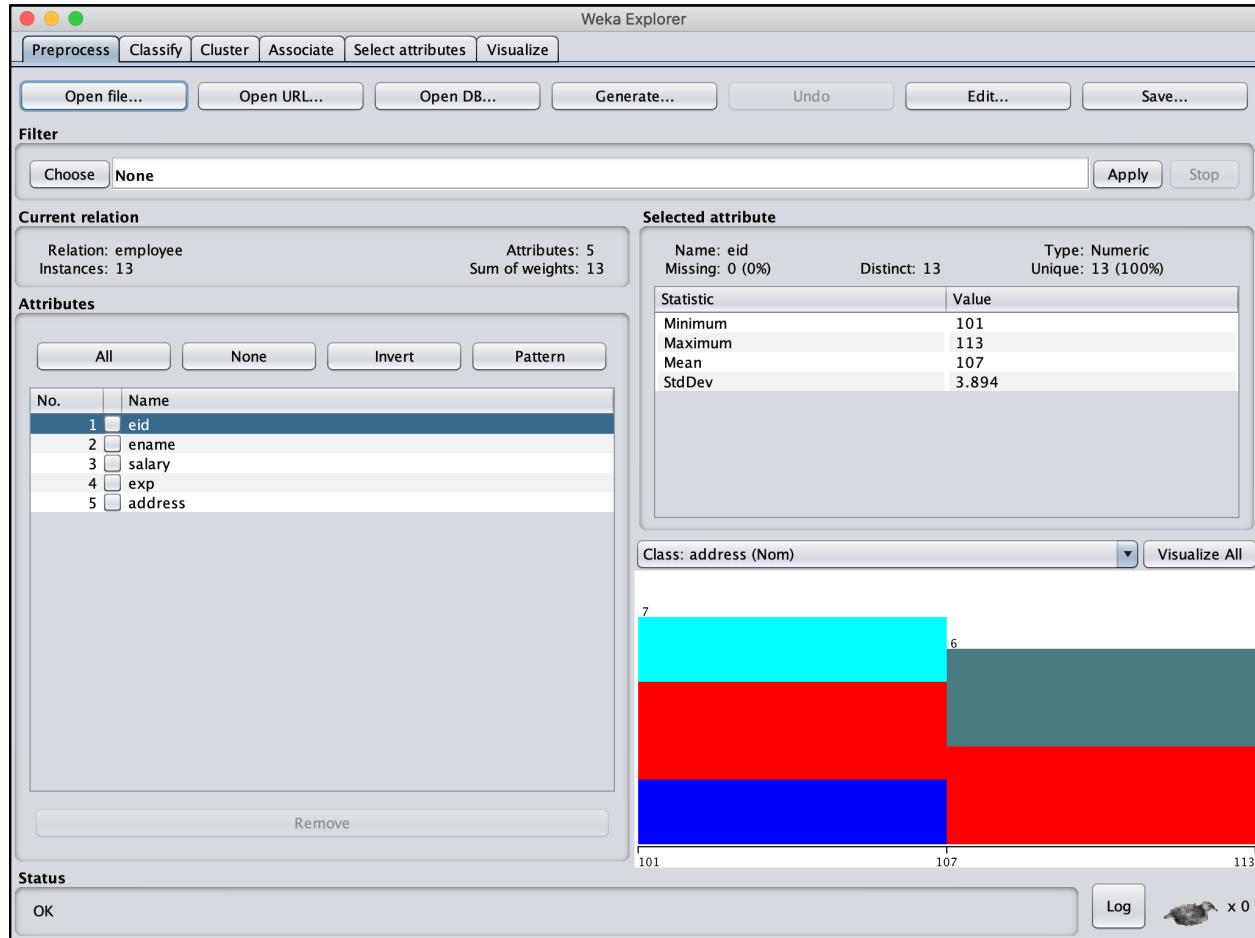
#	0 (53%)
1	7 (47%)

Status

OK

Practical 11 : Write a procedure for Employee data using Make Density Based Cluster Algorithm.

1. Data Loading



2. Review of Loaded Data

Viewer

Relation: employee

No.	1: eid	2: ename	3: salary	4: exp	5: address
	Numeric	Nominal	Numeric	Numeric	Nominal
1	10...	raj	1000...	4.0	pdtr
2	10...	ramu	1500...	5.0	pdtr
3	10...	anil	1200...	3.0	kdp
4	10...	sunil	1300...	3.0	kdp
5	10...	rajiv	1600...	6.0	kdp
6	10...	sunitha	1500...	5.0	nlr
7	10...	kavitha	1200...	3.0	nlr
8	10...	suresh	1100...	5.0	gtr
9	10...	ravi	1200...	3.0	gtr
...	11...	ramana	1100...	5.0	gtr
...	11...	ram	1200...	3.0	kdp
...	11...	kavya	1300...	4.0	kdp
...	11...	navya	1400...	5.0	kdp

Add instance Undo OK Cancel

3. Result after Make Density Based Cluster Algorithm.

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Clusterer

Choose SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 5

Cluster mode

- Use training set
- Supplied test set Set...
- Percentage split % 66
- Classes to clusters evaluation (Nom) class
- Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 00:55:48 – MakeDensityBasedClusterer
- 01:05:15 – SimpleKMeans

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 "weka.core.EuclideanDistance -R first-last" -I 5
Relation: customer
Instances: 10
Attributes: 4
name
age
income
class
Test mode: evaluate on training data

== Clustering model (full training set) ==
kMeans
=====
Number of iterations: 2
Within cluster sum of squared errors: 14.00000000000004

Initial starting points (random):
Cluster 0: u,middle,low,B
Cluster 1: w,youth,high,A

Missing values globally replaced with mean/mode

Final cluster centroids:
Attribute Full Data Cluster#
          (10.0)      0      1
=====
name           x        y        x
age          youth    youth    youth
income        high     low     high
class         A        B        A
```

Status

OK Log x 0

OK Log x 0

Final cluster centroids:

Attribute	Full Data	Cluster#	
	(10.0)	0	1
name	x	y	x
age	youth	youth	youth
income	high	low	high
class	A	B	A

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

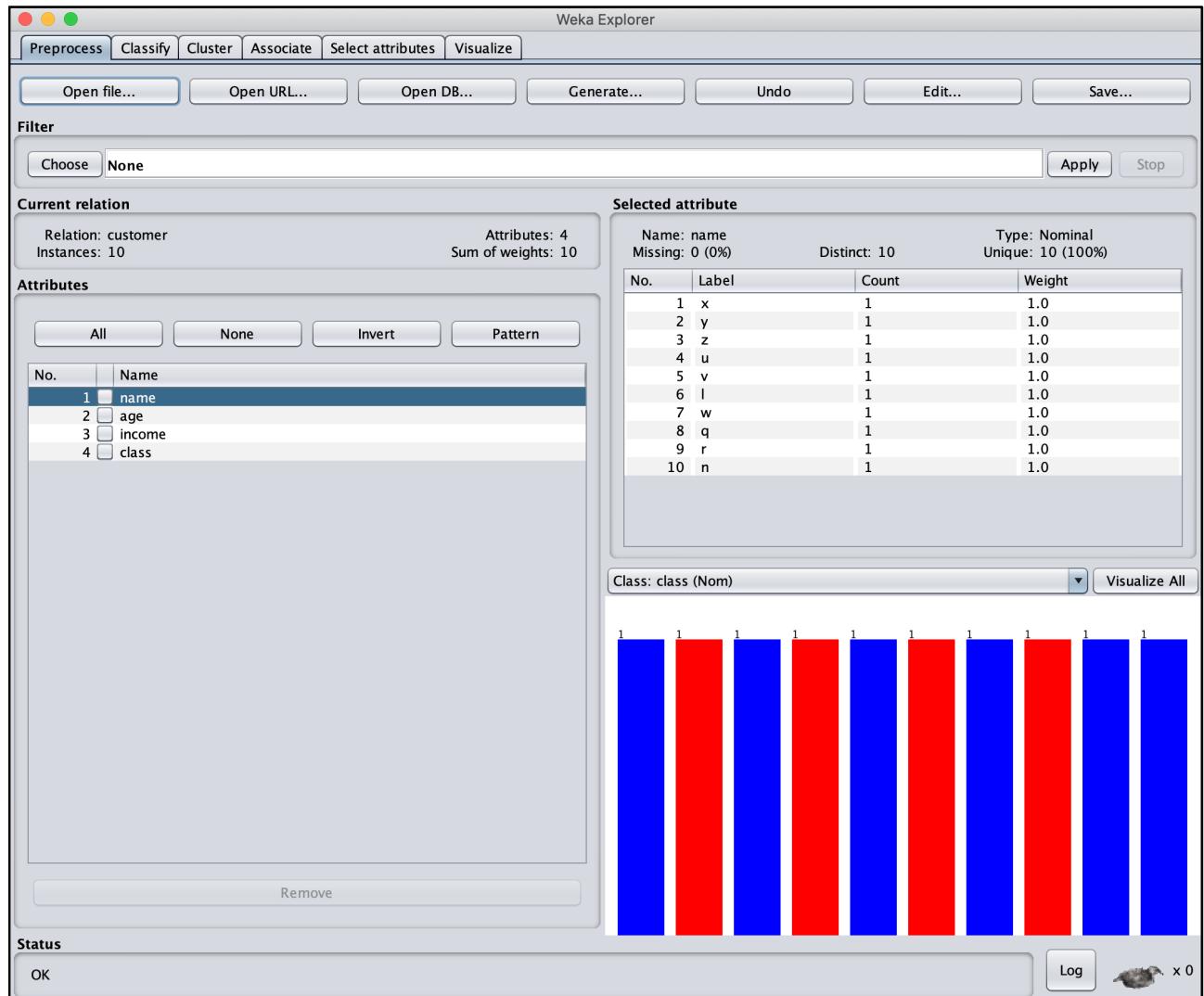
== Model and evaluation on training set ==

Clustered Instances

0	4 (40%)
1	6 (60%)

Practical 12 : Write a procedure for Clustering Customer data using Simple KMeans Algorithm.

1. Data Loading



2. Review of Loaded Data

The screenshot shows a software interface titled "Viewer". At the top left are three colored circles: red, grey, and green. The main title "Viewer" is centered above a section labeled "Relation: customer". Below this is a table with the following data:

No.	1: name	2: age	3: income	4: class
	Nominal	Nominal	Nominal	Nominal
1	x	youth	high	A
2	y	youth	low	B
3	z	mid...	high	A
4	u	mid...	low	B
5	v	senior	high	A
6	l	senior	low	B
7	w	youth	high	A
8	q	youth	low	B
9	r	mid...	high	A
...	n	senior	high	A

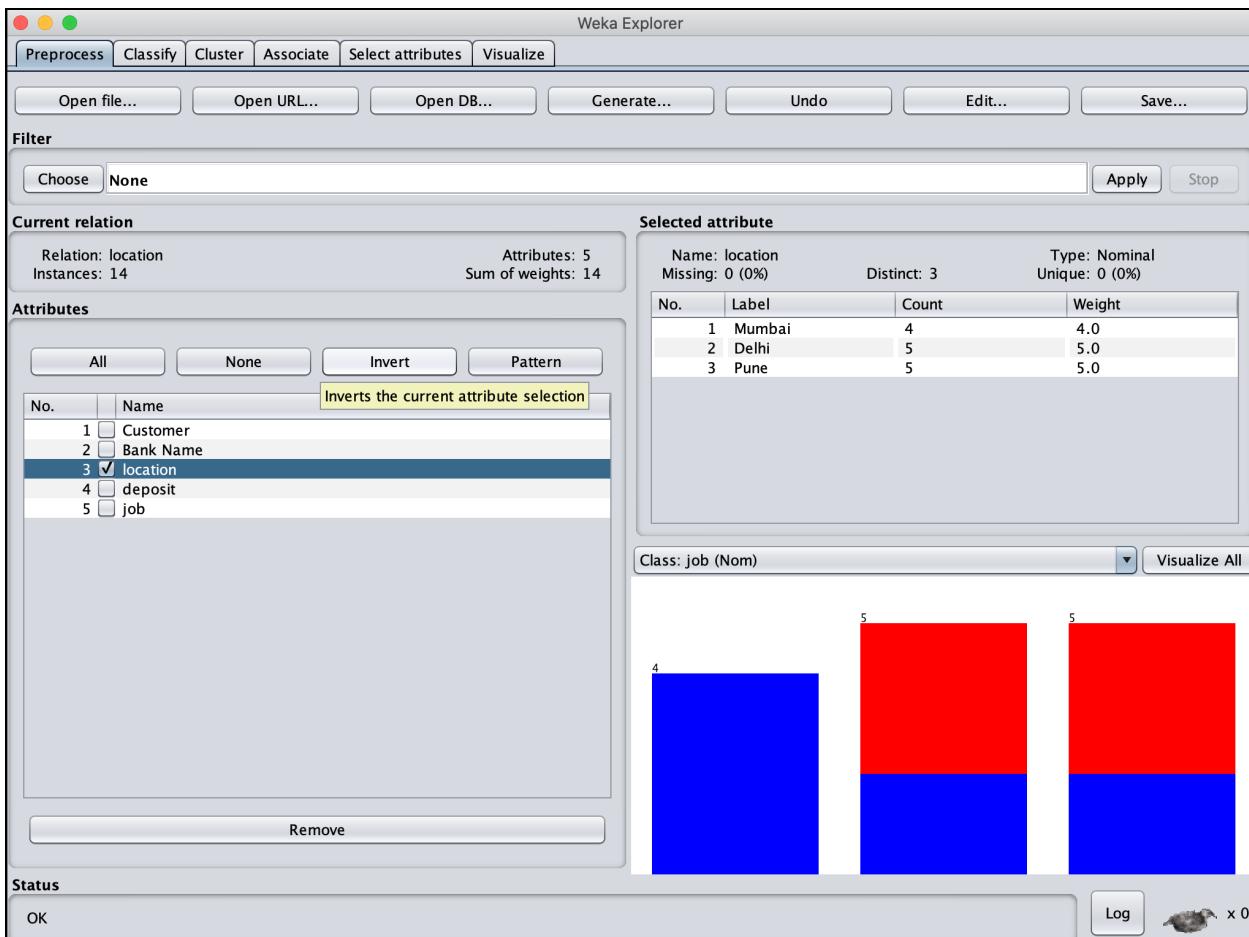
At the bottom of the window are three buttons: "Add instance", "Undo", and "OK".

3. Result after Simple KMeans Algorithm.

Practical 13: Decision Tree for Location Data (J48 classifier)

Dataset:

```
@relation location
@attribute Customer {m,f}
@attribute 'Bank Name' {Kotak,ICICI,LIC}
@attribute location {Mumbai,Delhi,Pune}
@attribute deposit {yes,no}
@attribute job {employed,unemployed}
@data
m,Kotak,Mumbai,yes,employed
m,ICICI,Delhi,yes,employed
f,ICICI,Pune,no,unemployed
m,LIC,Pune,yes,employed
f,Kotak,Mumbai,yes,employed
f,Kotak,Mumbai,no,employed
f,LIC,Delhi,yes,unemployed
m,ICICI,Pune,yes,unemployed
f,LIC,Delhi,yes,unemployed
m,Kotak,Delhi,no,employed
m,LIC,Pune,yes,employed
f,Kotak,Delhi,yes,unemployed
m,ICICI,Mumbai,no,employed
f,ICICI,Pune,yes,unemployed
```



Viewer

Relation: location

No. 1: Customer 2: Bank Name 3: location 4: deposit 5: job

No.	1: Customer	2: Bank Name	3: location	4: deposit	5: job
	Nominal	Nominal	Nominal	Nominal	Nominal
1	m	Kotak	Mumbai	yes	em...
2	m	ICICI	Delhi	yes	em...
3	f	ICICI	Pune	no	une...
4	m	LIC	Pune	yes	em...
5	f	Kotak	Mumbai	yes	em...
6	f	Kotak	Mumbai	no	em...
7	f	LIC	Delhi	yes	une...
8	m	ICICI	Pune	yes	une...
9	f	LIC	Delhi	yes	une...
...	m	Kotak	Delhi	no	em...
...	m	LIC	Pune	yes	em...
...	f	Kotak	Delhi	yes	une...
...	m	ICICI	Mumbai	no	em...
...	f	ICICI	Pune	yes	une...

Add instance Undo OK Cancel

After applying J48 classifier:

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) location ▾

Start Stop

Result list (right-click for options)

```
12:03:14 - treesJ48
12:22:12 - treesJ48
```

Classifier output

```
| Bank Name = Kotak: Mumbai (4.0/1.0)
| Bank Name = ICICI: Mumbai (2.0/1.0)
| Bank Name = LIC: Pune (2.0)
job = unemployed
| Bank Name = Kotak: Delhi (1.0)
| Bank Name = ICICI: Pune (3.0)
| Bank Name = LIC: Delhi (2.0)

Number of Leaves : 6
Size of the tree : 9

Time taken to build model: 0 seconds
==== Stratified cross-validation ====
==== Summary ====
Correctly Classified Instances      5          35.7143 %
Incorrectly Classified Instances    9          64.2857 %
Kappa statistic                   0.0455
Mean absolute error                0.407
Root mean squared error            0.5532
Relative absolute error             87.8933 %
Root relative squared error        112.2448 %
Total Number of Instances          14

==== Detailed Accuracy By Class ====


|               | TP Rate | FP Rate | Precision | Recall | F-Measure | MCC   | ROC Area | PRC Area | Class |
|---------------|---------|---------|-----------|--------|-----------|-------|----------|----------|-------|
| 0.500         | 0.400   | 0.333   | 0.500     | 0.400  | 0.091     | 0.525 | 0.310    | Mumbai   |       |
| 0.000         | 0.111   | 0.000   | 0.000     | 0.000  | -0.207    | 0.522 | 0.443    | Delhi    |       |
| 0.600         | 0.444   | 0.429   | 0.600     | 0.500  | 0.149     | 0.667 | 0.542    | Pune     |       |
| Weighted Avg. | 0.357   | 0.313   | 0.248     | 0.357  | 0.293     | 0.005 | 0.575    | 0.440    |       |


==== Confusion Matrix ====


```
a b c <-- classified as
2 1 1 | a = Mumbai
2 0 3 | b = Delhi
2 0 3 | c = Pune
```



Status



OK Log x 0


```

Decision Tree

