Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

A.Y. 2021-2022

LAB EXPERIMENT NO. 01

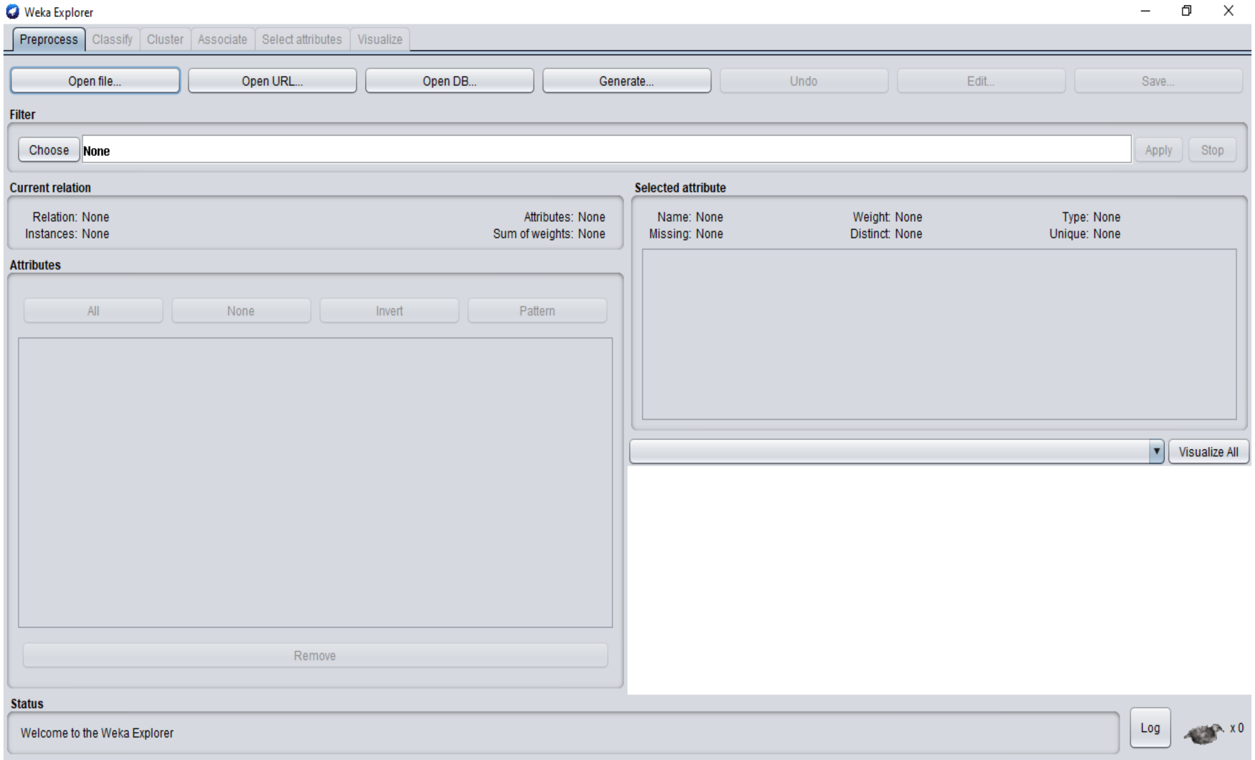
Aim: Perform data Pre-processing task using Weka data mining tool

Theory:

WEKA - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems

Tasks performed through Weka:

Preprocessing:
Classification:
Clustering:
Association Rule:
Select Attributes:
Visualization:



Filter: None

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

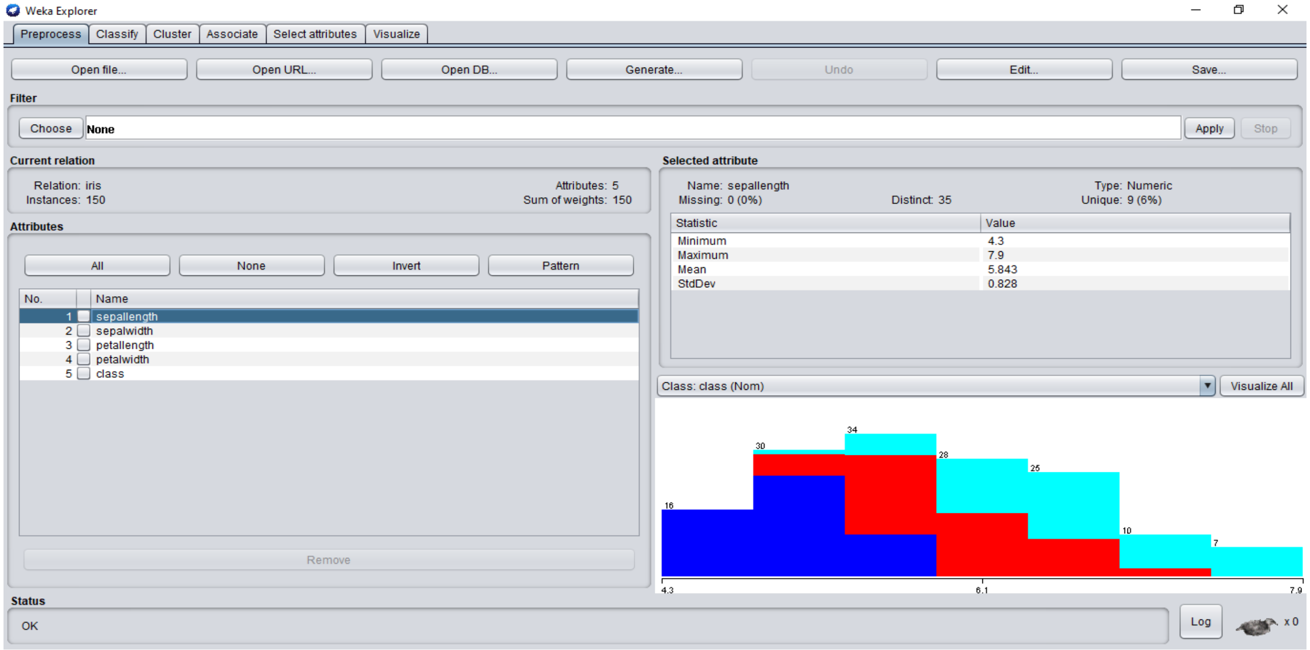
Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Weka GUI

Preprocessing activities to be observed in Weka:

1. **Visualization:** Visualize scatter plot for all the attributes from dataset selected from Weka. Determine correlation if any using these plots for different datasets



Filter: None

Current relation

Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150

Selected attribute

Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

Attributes

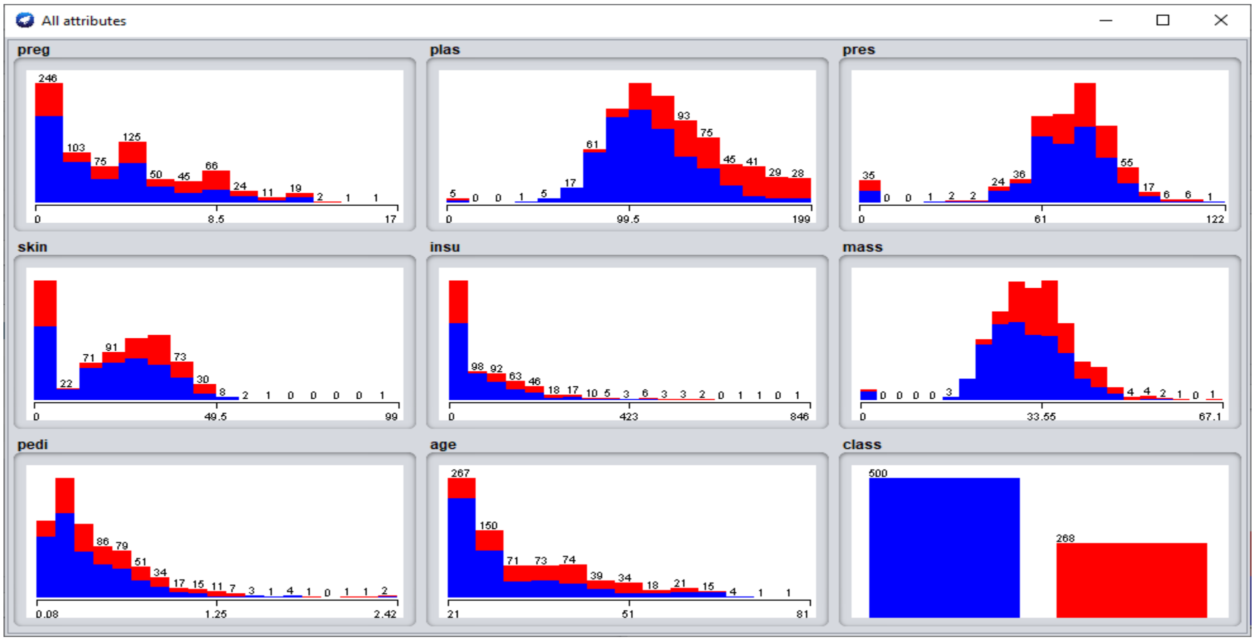
No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Class: class (Nom)

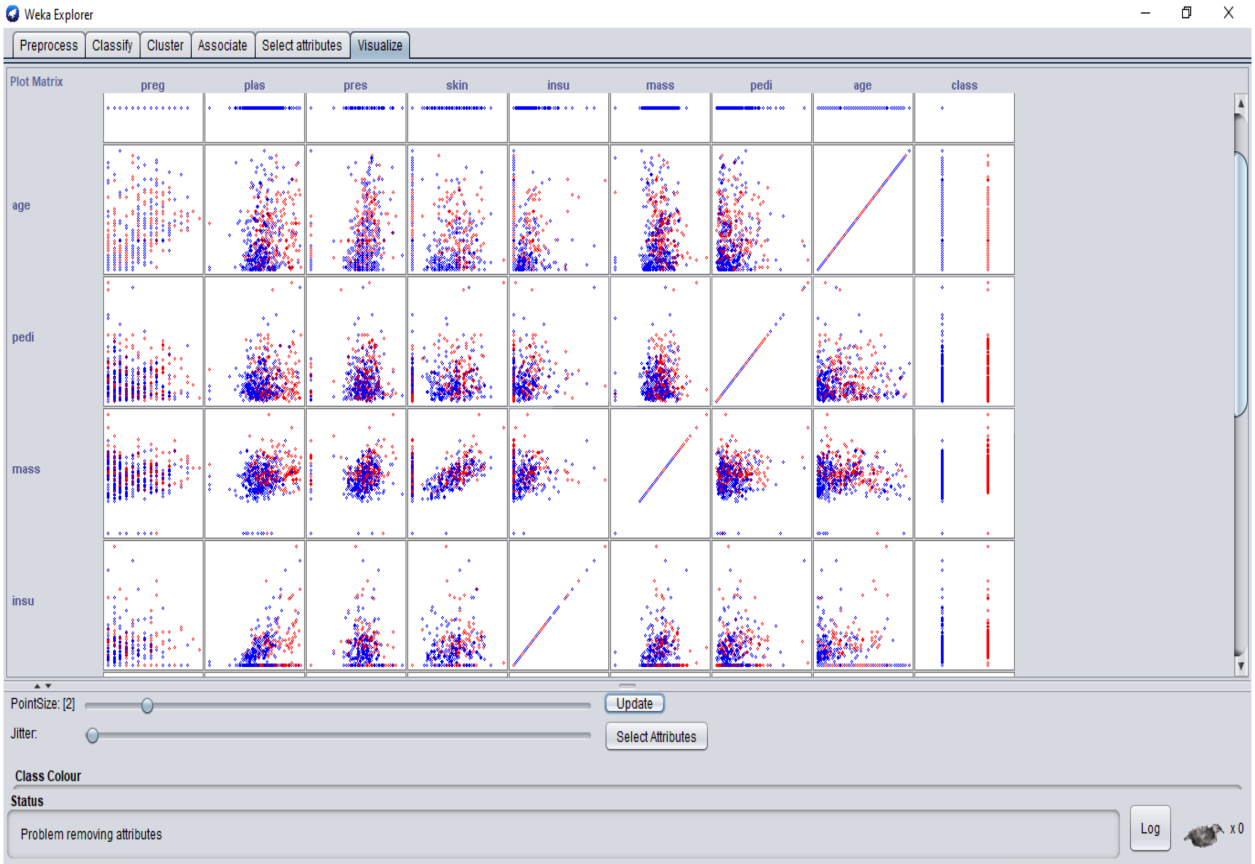
Status

OK

Weka on loading **Prima Diabetes dataset**



Visualize all – Distribution Plot



PointSize: [2]

Jitter:

Problem removing attributes

Correlation between the features

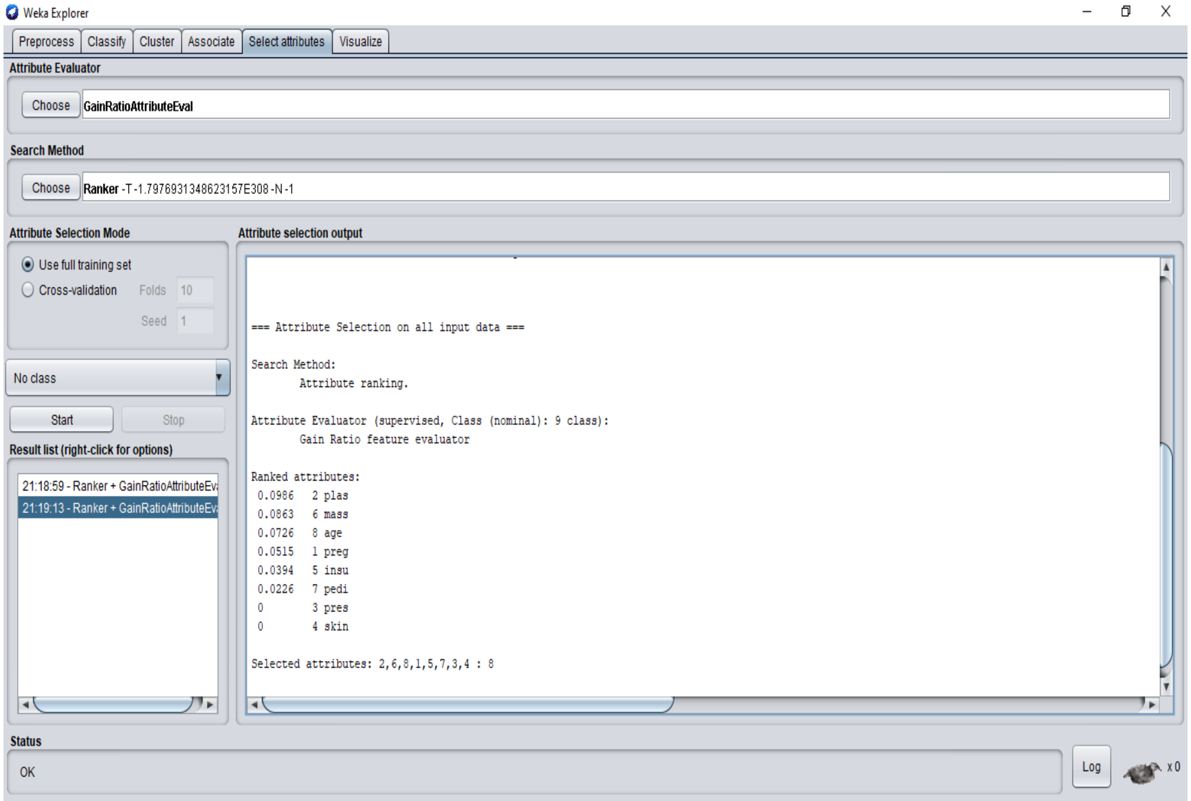
Thus, upon performing **data visualization** we observed :

i) The data distribution with respect to each features and the skewness of the data with respect to that feature.

ii) **Scatter plot** between the features. For e.g in the Prima Diabetes dataset:

- Age vs Pregnancy have no co-relation.
- Plasma vs insurance were positively correlated
- Mass vs skin was positively correlated.

2. **Select Attributes:** Apply suitable feature selection filter like GainRatio etc to choose relevant attributes from the list of attributes. Observe the ranks / priority provided by the filter.



Attribute Evaluator: GainRatioAttributeEval

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

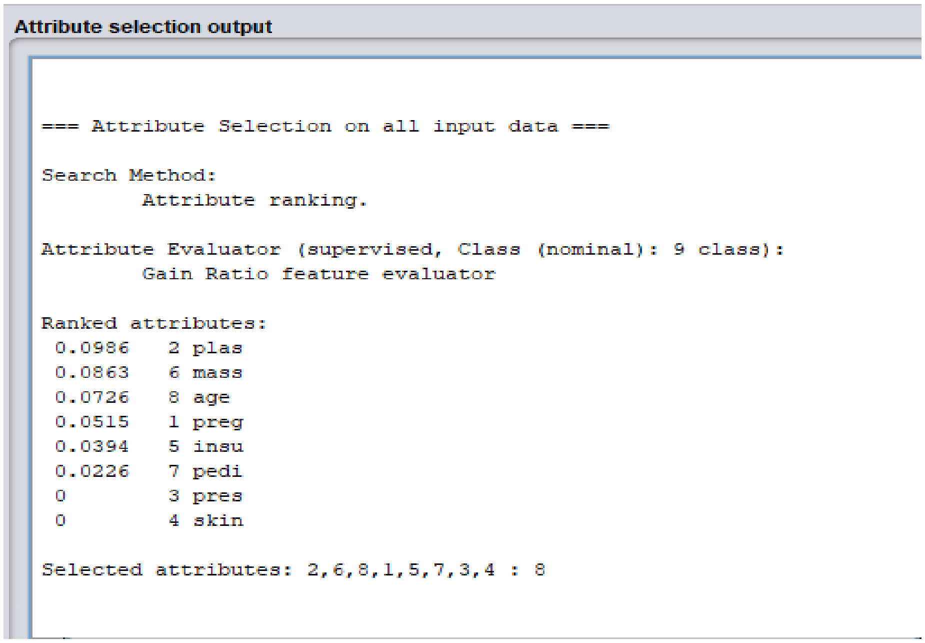
Attribute Selection Mode:
● Use full training set
○ Cross-validation Folds 10 Seed 1

Result list (right-click for options):
21:18:59 - Ranker + GainRatioAttributeEv
21:19:13 - Ranker + GainRatioAttributeEv

Attribute selection output:

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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.0986   2 plas
 0.0863   6 mass
 0.0726   8 age
 0.0515   1 preg
 0.0394   5 insu
 0.0226   7 pedi
 0        3 pres
 0        4 skin

Selected attributes: 2,6,8,1,5,7,3,4 : 8
```

Status: OK



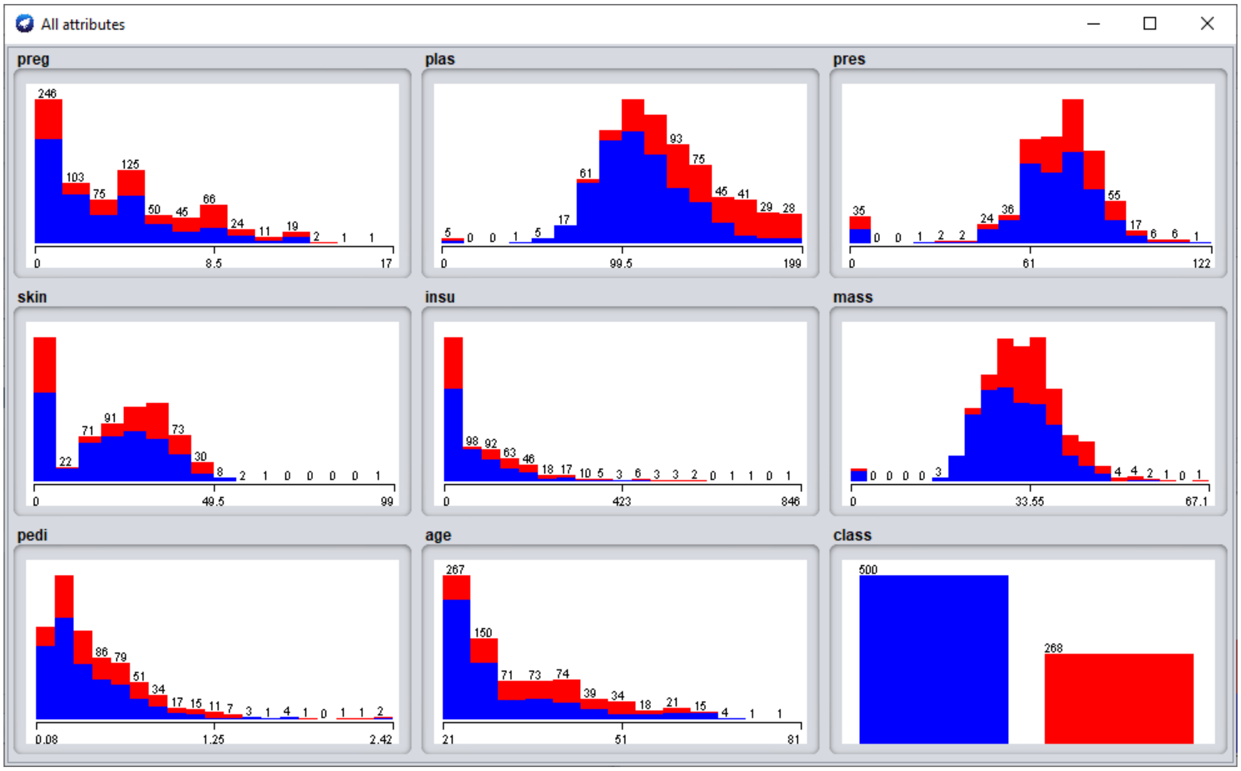
Attribute selection output

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

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 9 class):
        Gain Ratio feature evaluator

Ranked attributes:
 0.0986   2 plas
 0.0863   6 mass
 0.0726   8 age
 0.0515   1 preg
 0.0394   5 insu
 0.0226   7 pedi
 0        3 pres
 0        4 skin

Selected attributes: 2,6,8,1,5,7,3,4 : 8
```

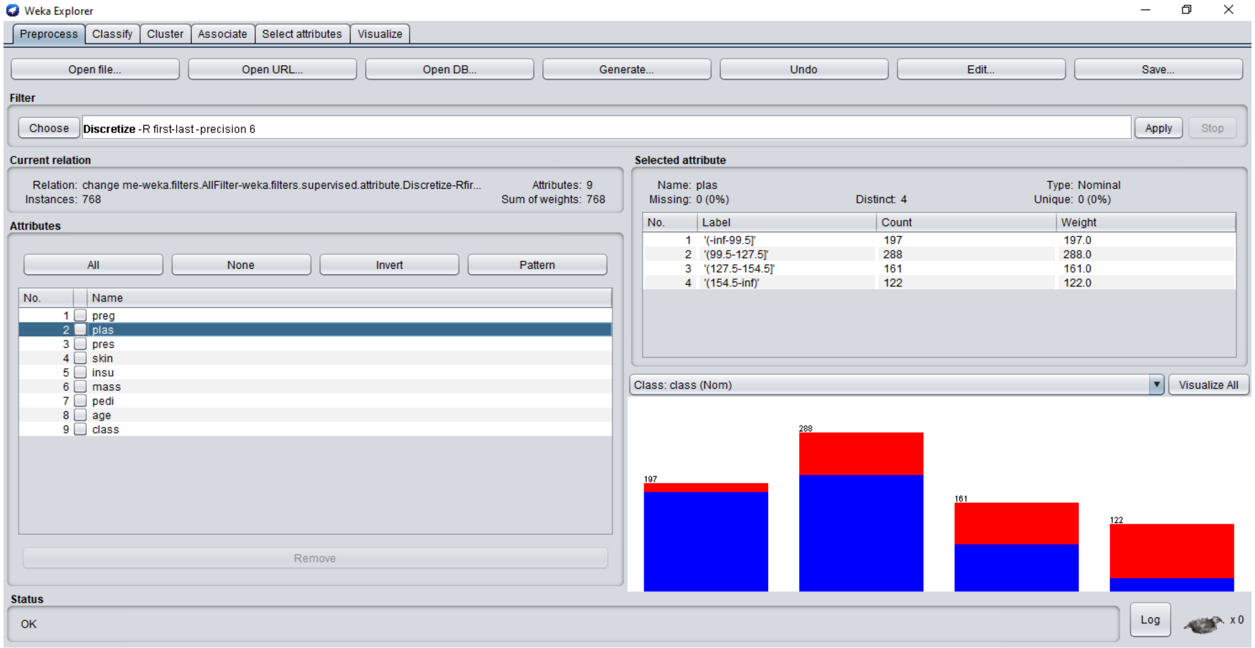
Using attribute evaluator as InfoGainAttributeEval on Ranker Search method in the Select Attribute tab we got the order and values of the most important attributes through entropy which further is used for clusters/classification. Thus, from this we know that attribute ‘plas’ , ‘mass’ then ‘age’ and so on hold importance while clustering the instances.

3. Preprocessing:

a. **Visualize All:** Select this button to visualize histograms of all attributes.



b. Filter: Choose Discretization under Unsupervised and Supervised methods. Observe the discretization and the outliers.



Filter: **Discretize** -R first-last -precision 6

Current relation

Relation: change me-weka.filters.AllFilter-weka.filters.supervised.attribute.Discretize-Rfir... Attributes: 9
Instances: 768 Sum of weights: 768

Attributes

No.	Name
1	preg
2	plas
3	pres
4	skin
5	insu
6	mass
7	pedi
8	age
9	class

Selected attribute

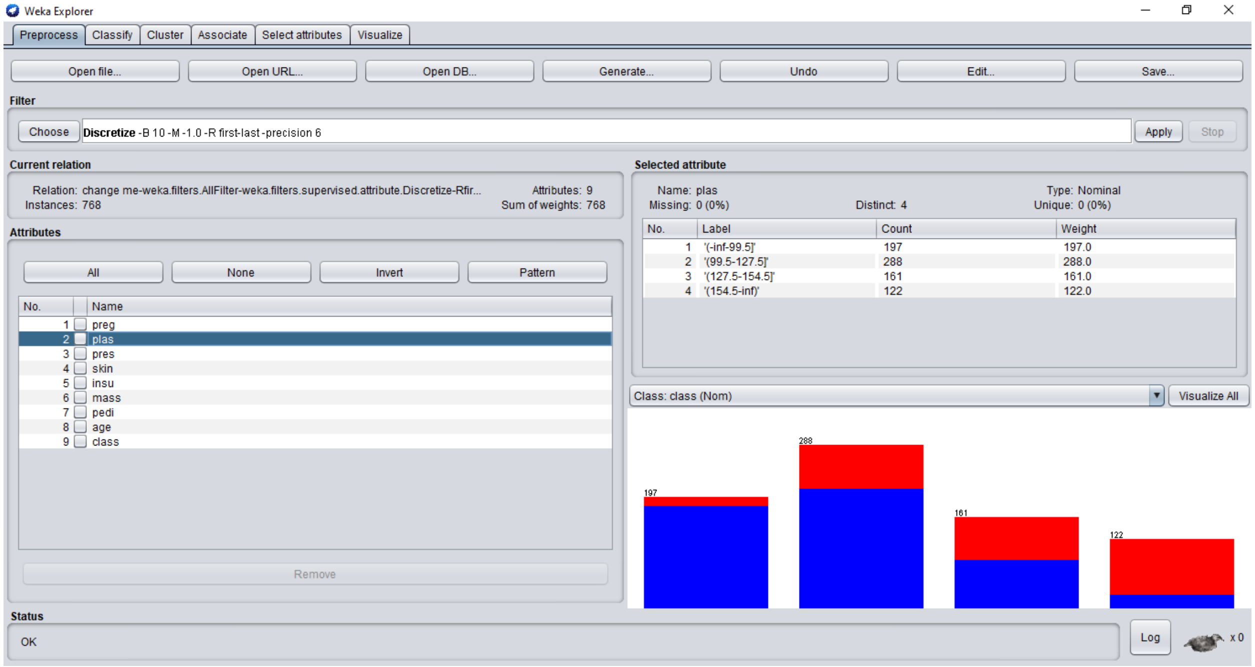
Name: plas Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-99.5]'	197	197.0
2	'(99.5-127.5]'	288	288.0
3	'(127.5-154.5]'	161	161.0
4	'(154.5-inf)'	122	122.0

Class: class (Nom)

Status
OK

Filter – Supervised Discretize



Filter: **Discretize** -B 10 -M -1.0 -R first-last -precision 6

Current relation

Relation: change me-weka.filters.AllFilter-weka.filters.supervised.attribute.Discretize-Rfir... Attributes: 9
Instances: 768 Sum of weights: 768

Attributes

No.	Name
1	preg
2	plas
3	pres
4	skin
5	insu
6	mass
7	pedi
8	age
9	class

Selected attribute

Name: plas Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

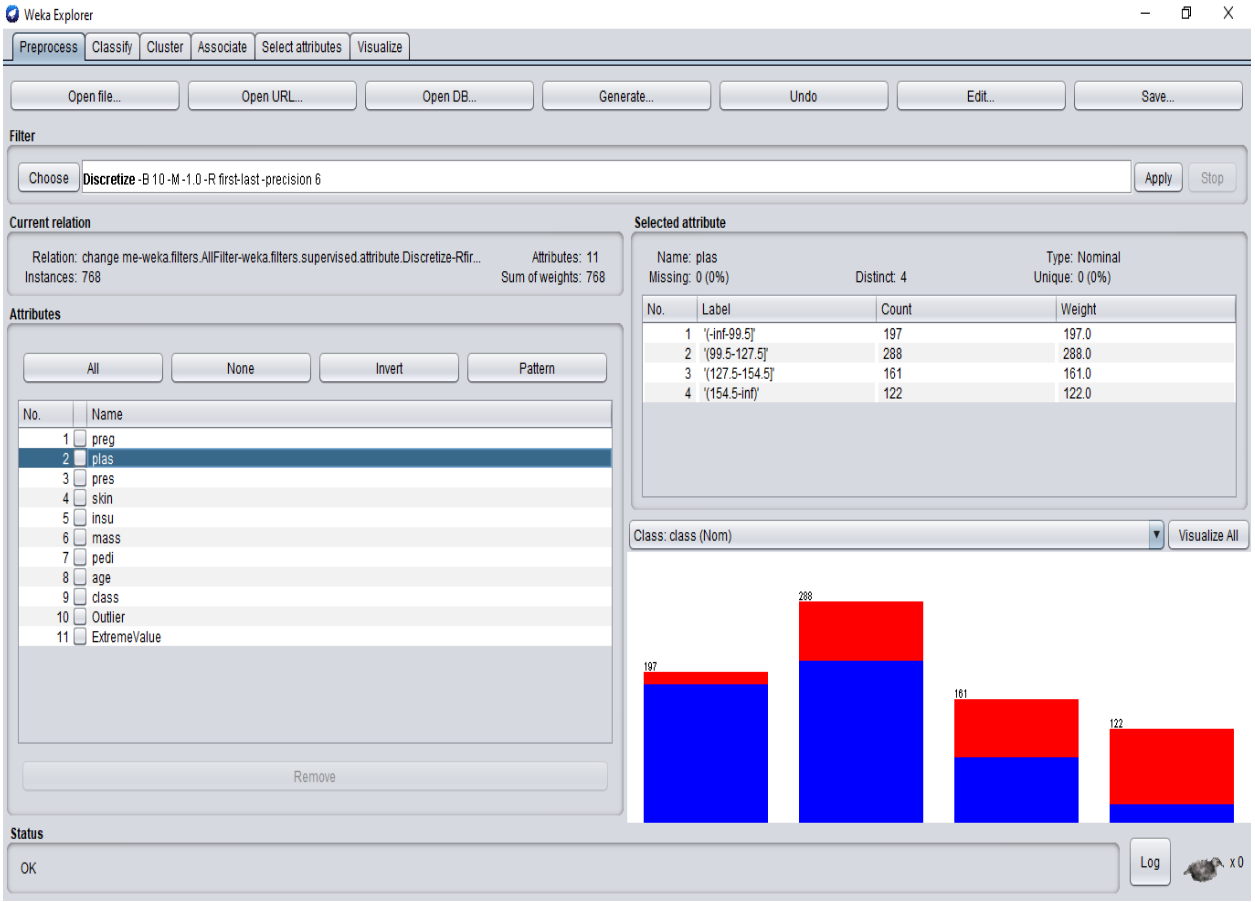
No.	Label	Count	Weight
1	'(-inf-99.5]'	197	197.0
2	'(99.5-127.5]'	288	288.0
3	'(127.5-154.5]'	161	161.0
4	'(154.5-inf)'	122	122.0

Class: class (Nom)

Status
OK

Filter – Unsupervised Discretize

c. **IQR:** Observe the IQR values for a selected attribute. Observe the outlier and extreme values



Filter

Discretize -B 10 -M -1.0 -R first-last -precision 6

Current relation

Relation: change me-weka.filters.AllFilter-weka.filters.supervised.attribute.Discretize-Rfir... Attributes: 11
Instances: 768 Sum of weights: 768

Attributes

No.		Name
1	□	preg
2	□	plas
3	□	pres
4	□	skin
5	□	insu
6	□	mass
7	□	pedi
8	□	age
9	□	class
10	□	Outlier
11	□	ExtremeValue

Selected attribute

Name: plas Type: Nominal
Missing: 0 (0%) Distinct: 4 Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-99.5]'	197	197.0
2	'(99.5-127.5]'	288	288.0
3	'(127.5-154.5]'	161	161.0
4	'(154.5-inf)'	122	122.0

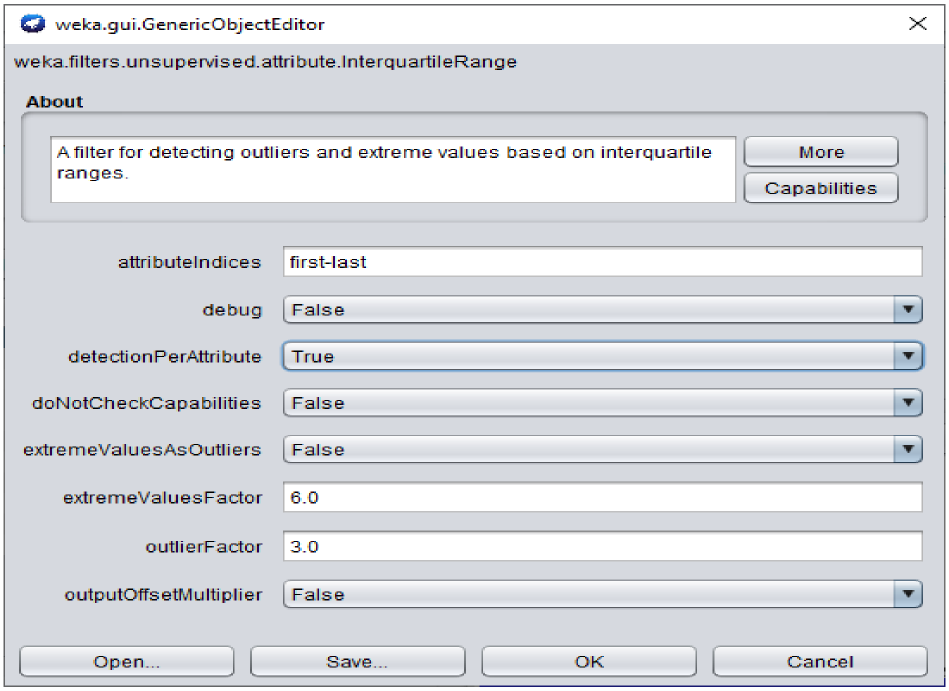
Class: class (Nom)

Status

OK

Filter – IOR

Thus, by using IOR filter we can look at the outliers i.e. those values which are outside the 1.5*IOR range. Data cleaning is necessary as these extreme outliers do affect the model accuracy.


weka.gui.GenericObjectEditor

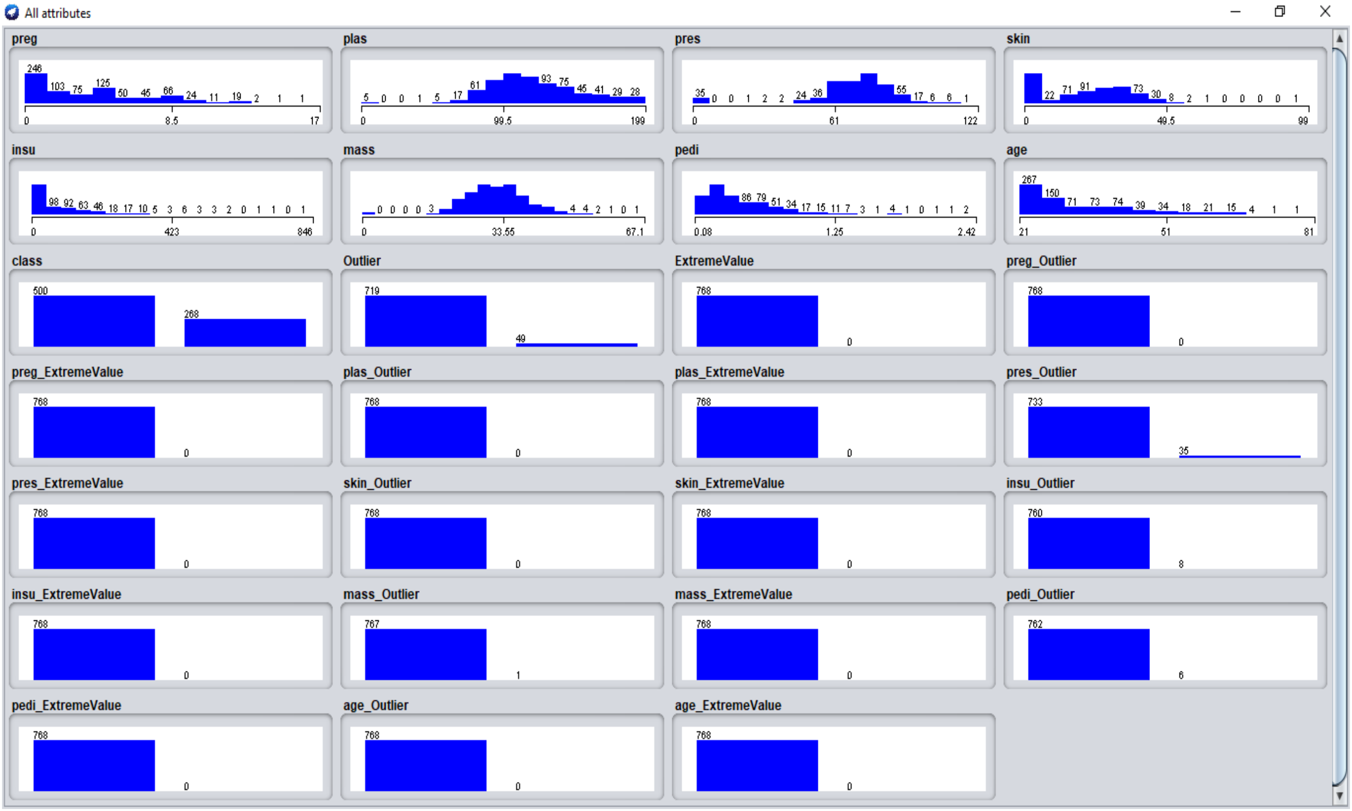
weka.filters.unsupervised.attribute.InterquartileRange

About

A filter for detecting outliers and extreme values based on interquartile ranges.

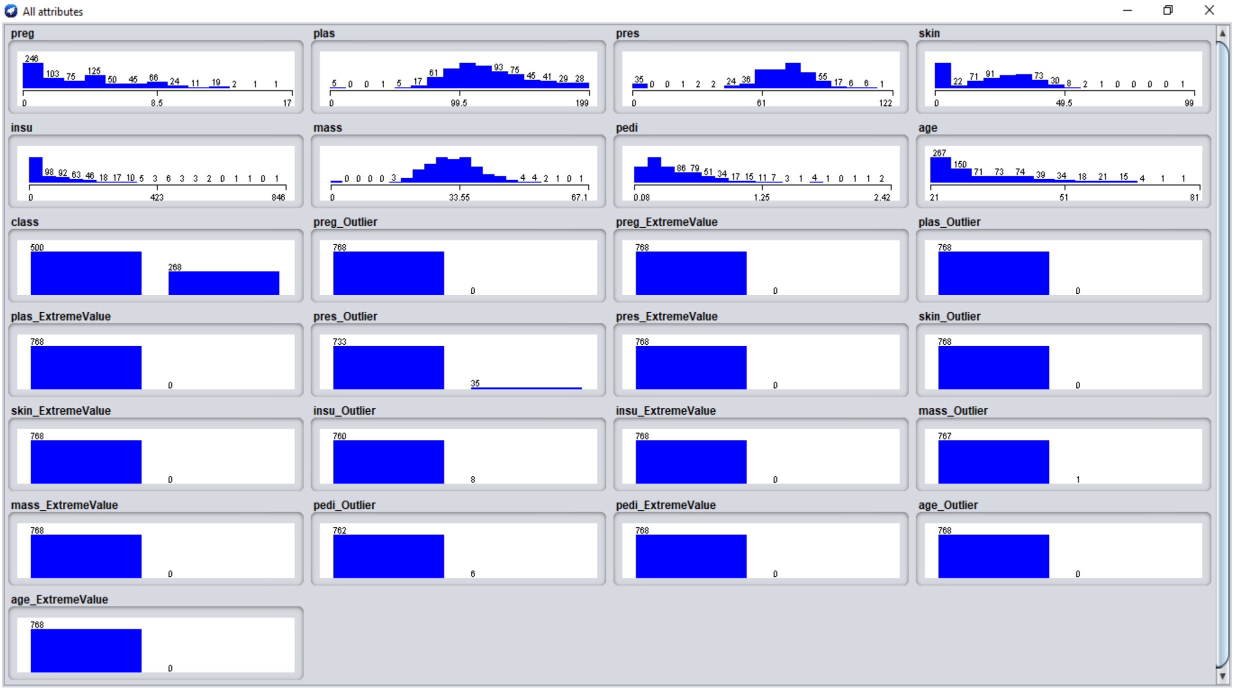
attributeIndices	first-last
debug	False
detectionPerAttribute	True
doNotCheckCapabilities	False
extremeValuesAsOutliers	False
extremeValuesFactor	6.0
outlierFactor	3.0
outputOffsetMultiplier	False

IOR → detectPerAttribute – True

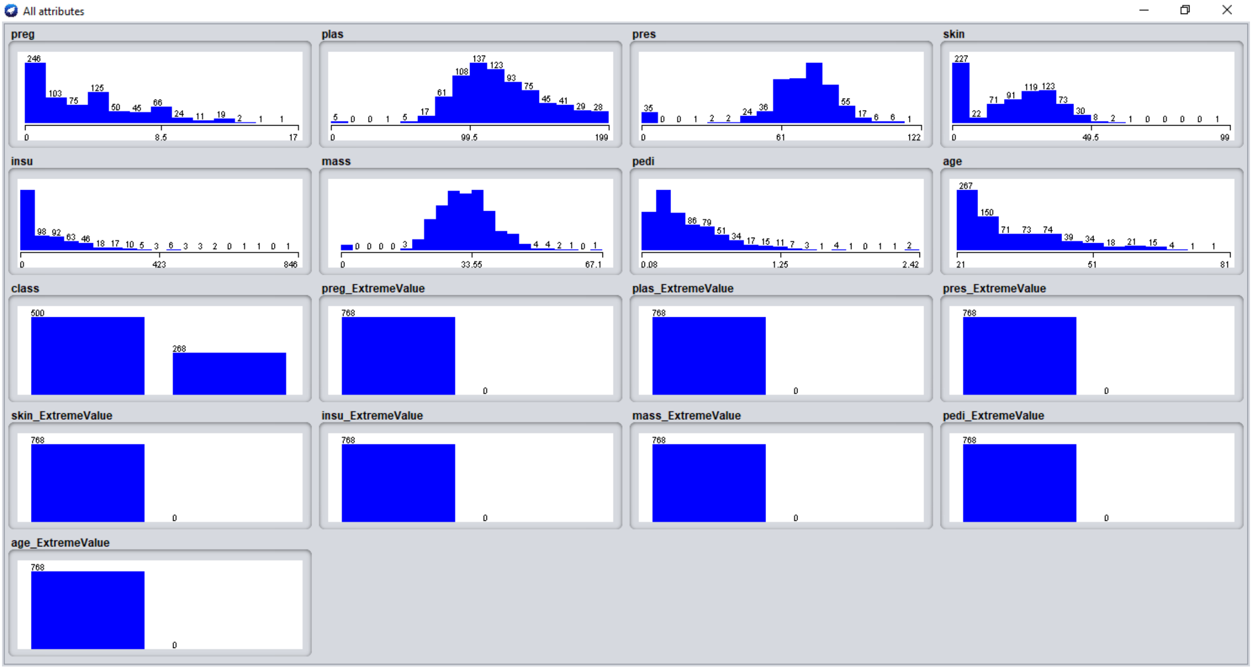


Visualize all – Outliers and Extreme Values

d. Removethevalue: Remove instances with outlier values and show the screenshots of dataset before and after the removal.

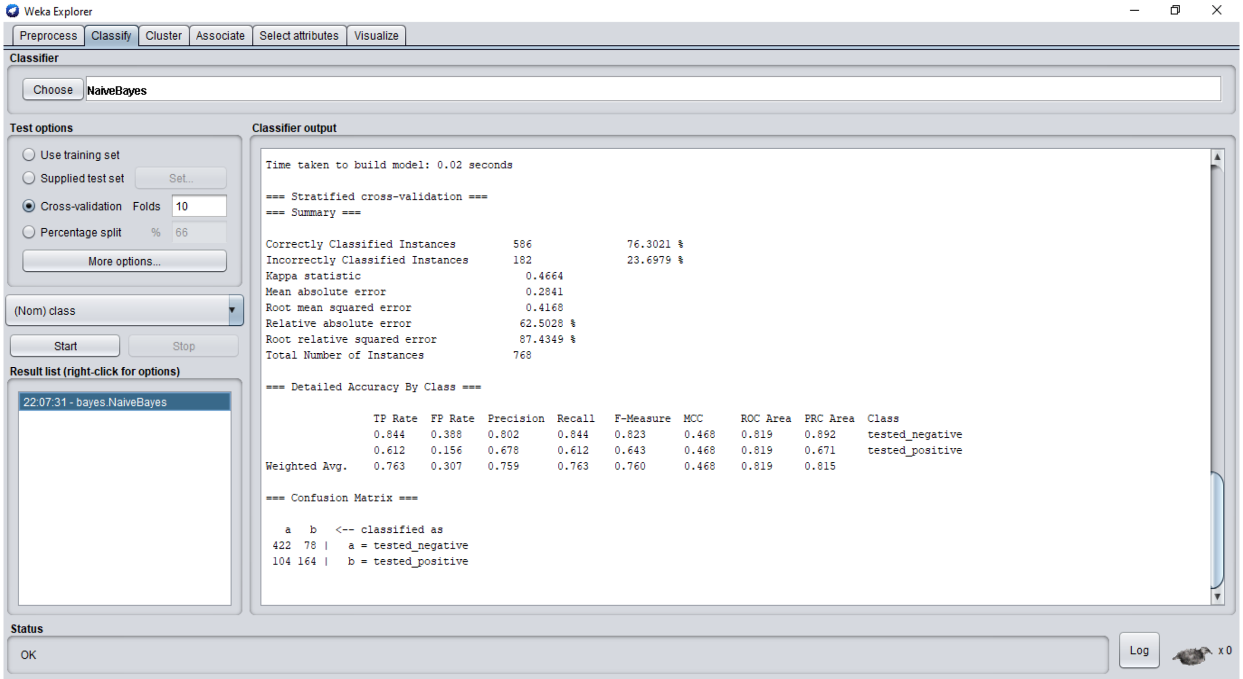


Before Removal of Outliers



After Removal of Outliers

4. **Classification:** Perform NB and Random Forest classification



Classifier: NaiveBayes

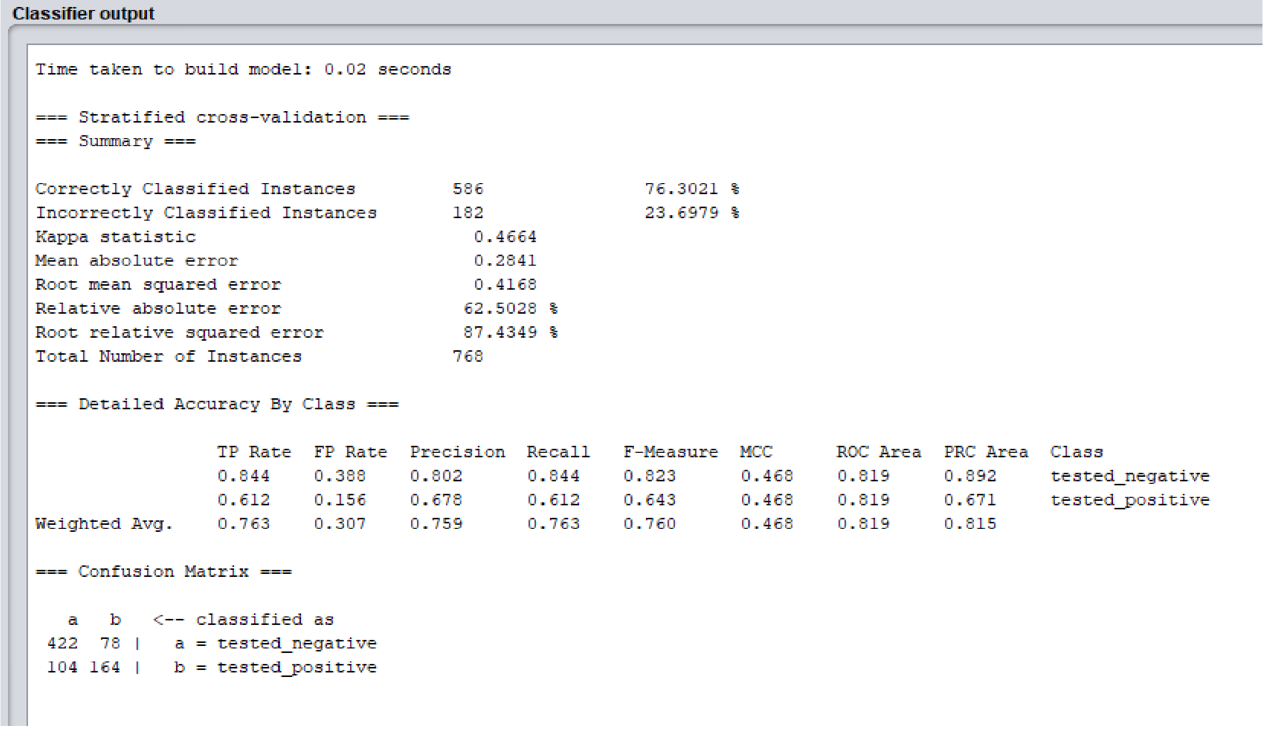
● Cross-validation Folds 10

(Nom) class

22:07:31 - bayes.NaiveBayes

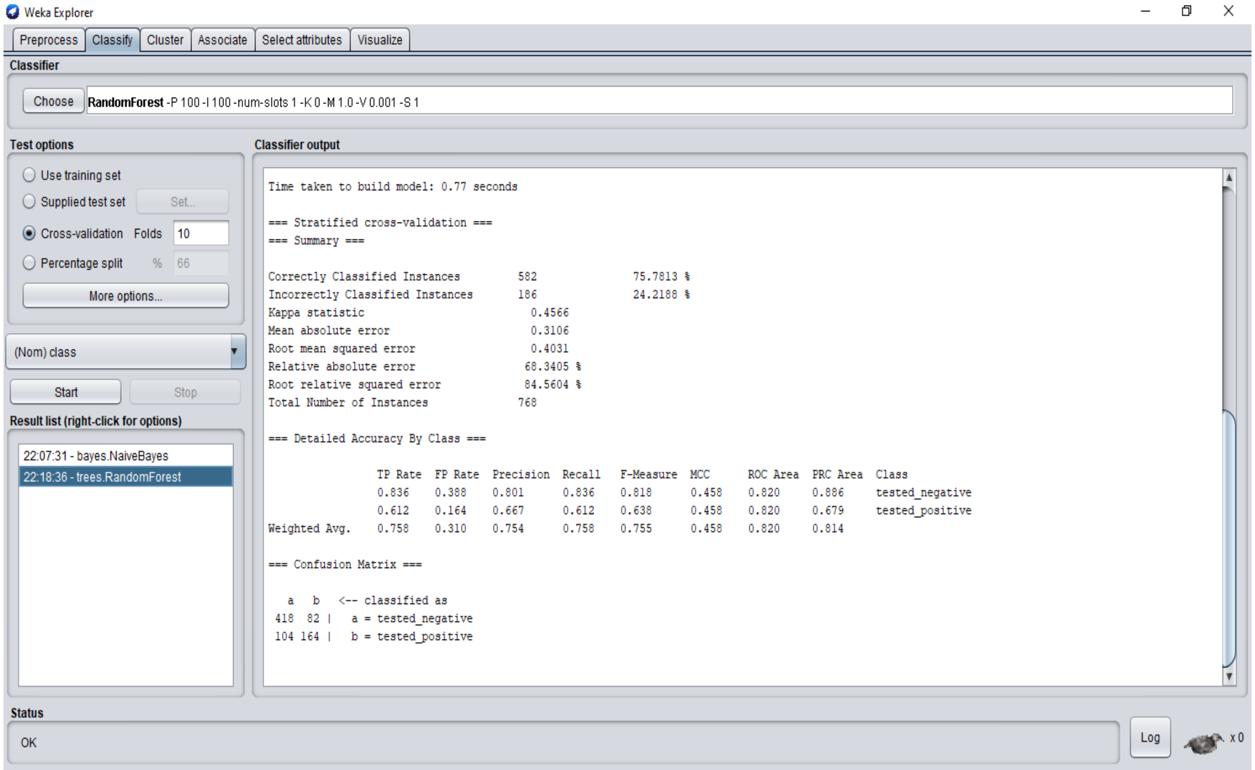
Status: OK

```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances         586               76.3021 %
Incorrectly Classified Instances       182               23.6979 %
Kappa statistic                          0.4664
Mean absolute error                      0.2841
Root mean squared error                  0.4168
Relative absolute error                 62.5028 %
Root relative squared error             87.4349 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.844    0.388    0.802      0.844    0.823      0.468    0.819     0.892     tested_negative
                 0.612    0.156    0.678      0.612    0.643      0.468    0.819     0.671     tested_positive
Weighted Avg.    0.763    0.307    0.759      0.763    0.760      0.468    0.819     0.815

=== Confusion Matrix ===

   a   b   <-- classified as
 422  78 |   a = tested_negative
 104 164 |   b = tested_positive
```



```
Time taken to build model: 0.02 seconds

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

Correctly Classified Instances         586               76.3021 %
Incorrectly Classified Instances       182               23.6979 %
Kappa statistic                          0.4664
Mean absolute error                      0.2841
Root mean squared error                  0.4168
Relative absolute error                 62.5028 %
Root relative squared error             87.4349 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.844    0.388    0.802      0.844    0.823      0.468    0.819     0.892     tested_negative
                 0.612    0.156    0.678      0.612    0.643      0.468    0.819     0.671     tested_positive
Weighted Avg.    0.763    0.307    0.759      0.763    0.760      0.468    0.819     0.815

=== Confusion Matrix ===

   a   b   <-- classified as
 422  78 |   a = tested_negative
 104 164 |   b = tested_positive
```

Naïve Bayes on pima diabetes



Classifier: RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

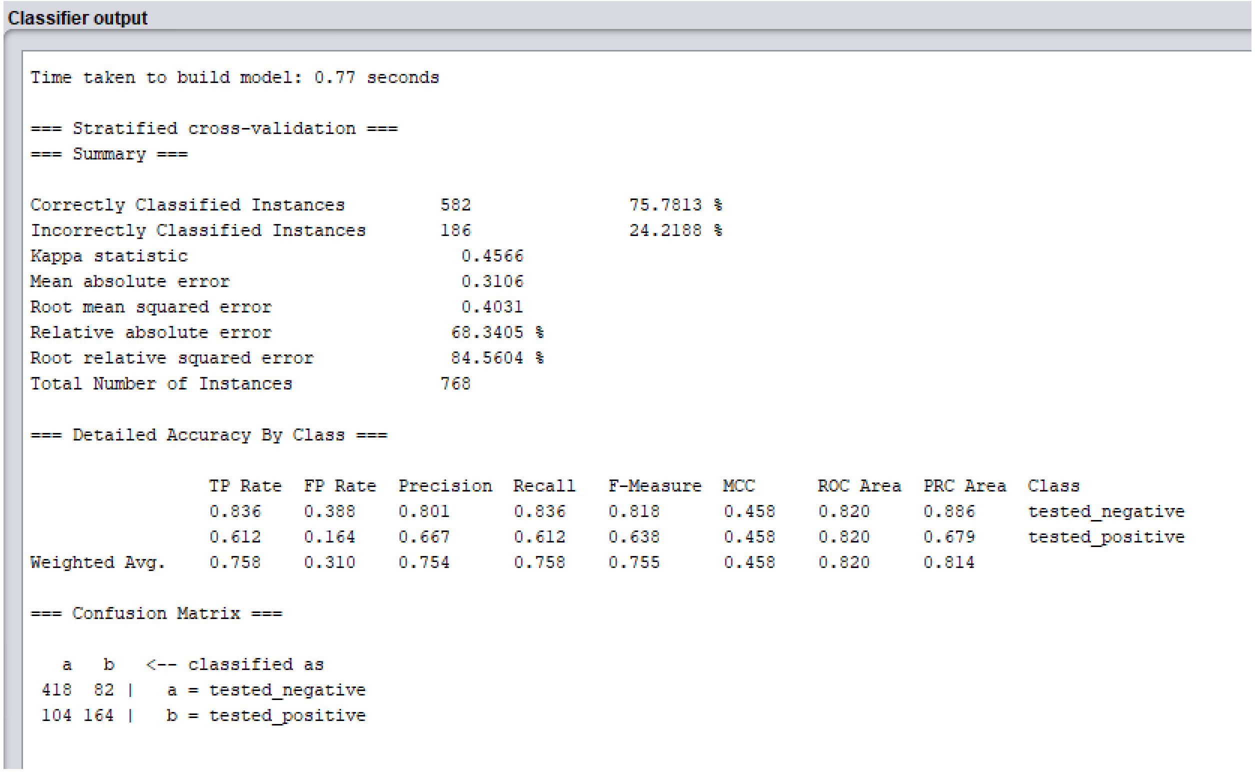
Test options:
- ○ Use training set
- ○ Supplied test set
- ● Cross-validation Folds 10
- ○ Percentage split % 66

(Nom) class

Result list (right-click for options):
- 22:07:31 - bayes.NaiveBayes
- 22:18:36 - trees.RandomForest

Status: OK

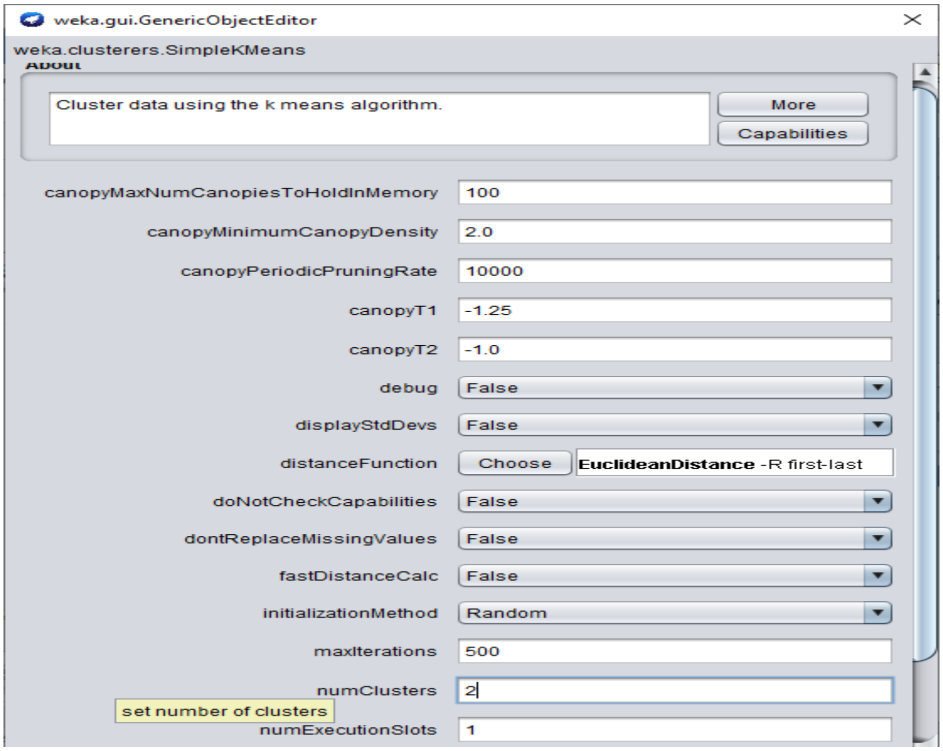
```
Time taken to build model: 0.77 seconds

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

Correctly Classified Instances         582               75.7813 %
Incorrectly Classified Instances       186               24.2188 %
Kappa statistic                          0.4566
Mean absolute error                      0.3106
Root mean squared error                  0.4031
Relative absolute error                 68.3405 %
Root relative squared error             84.5604 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.836    0.388    0.801      0.836    0.818      0.458    0.820     0.886     tested_negative
                 0.612    0.164    0.667      0.612    0.638      0.458    0.820     0.679     tested_positive
Weighted Avg.    0.758    0.310    0.754      0.758    0.755      0.458    0.820     0.814

=== Confusion Matrix ===

   a   b   <-- classified as
 418  82 |   a = tested_negative
 104 164 |   b = tested_positive
```



```
Time taken to build model: 0.77 seconds

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

Correctly Classified Instances         582               75.7813 %
Incorrectly Classified Instances       186               24.2188 %
Kappa statistic                          0.4566
Mean absolute error                      0.3106
Root mean squared error                  0.4031
Relative absolute error                 68.3405 %
Root relative squared error             84.5604 %
Total Number of Instances              768

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.836    0.388    0.801      0.836    0.818      0.458    0.820     0.886     tested_negative
                 0.612    0.164    0.667      0.612    0.638      0.458    0.820     0.679     tested_positive
Weighted Avg.    0.758    0.310    0.754      0.758    0.755      0.458    0.820     0.814

=== Confusion Matrix ===

   a   b   <-- classified as
 418  82 |   a = tested_negative
 104 164 |   b = tested_positive
```

Random Forest Classifier

5. **Clustering:** Perform kmeans, hierarchical clustering and explain the output



weka.gui.GenericObjectEditor

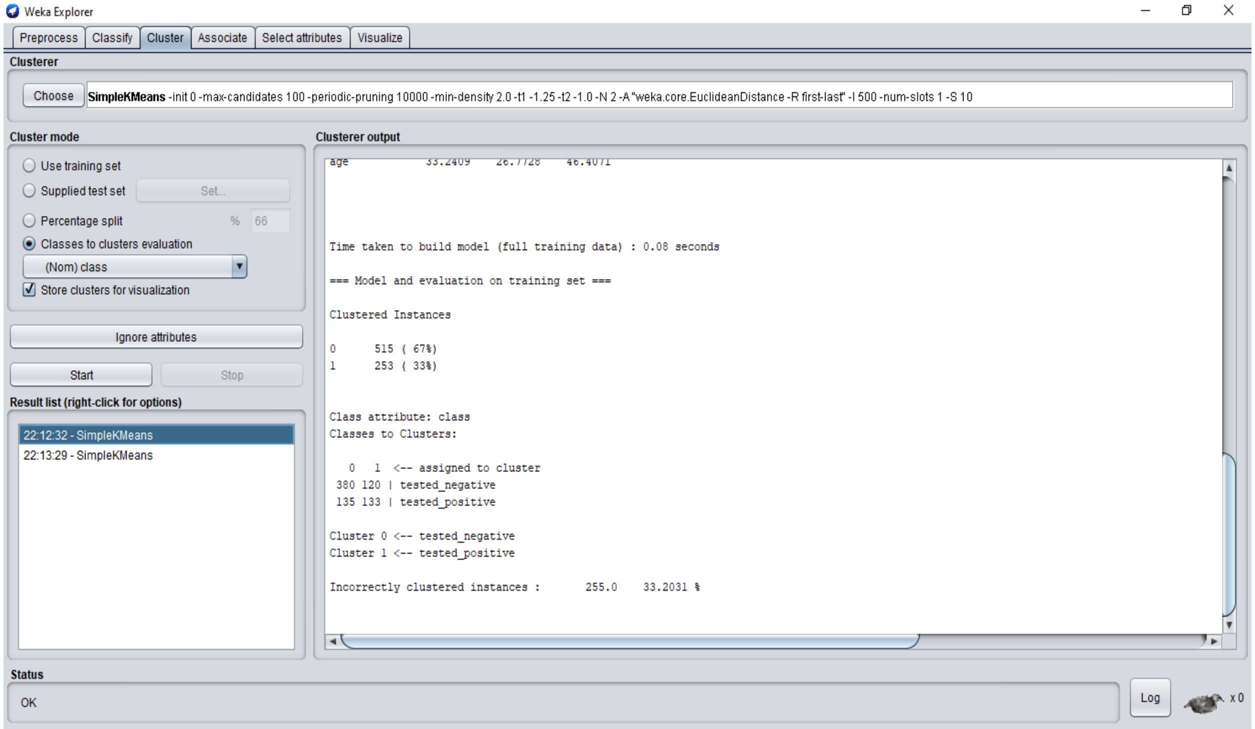
weka.clusterers.SimpleKMeans

About

Cluster data using the k means algorithm.

canopyMaxNumCanopiesToHoldInMemory: 100
canopyMinimumCanopyDensity: 2.0
canopyPeriodicPruningRate: 10000
canopyT1: -1.25
canopyT2: -1.0
debug: False
displayStdDevs: False
distanceFunction: Choose **EuclideanDistance** -R first-last
doNotCheckCapabilities: False
dontReplaceMissingValues: False
fastDistanceCalc: False
initializationMethod: Random
maxIterations: 500
numClusters: 2
numExecutionSlots: 1

set number of clusters



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 500 -num-slots 1 -S 10

Cluster mode

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

Result list (right-click for options)

22:12:32 - SimpleKMeans
22:13:29 - SimpleKMeans

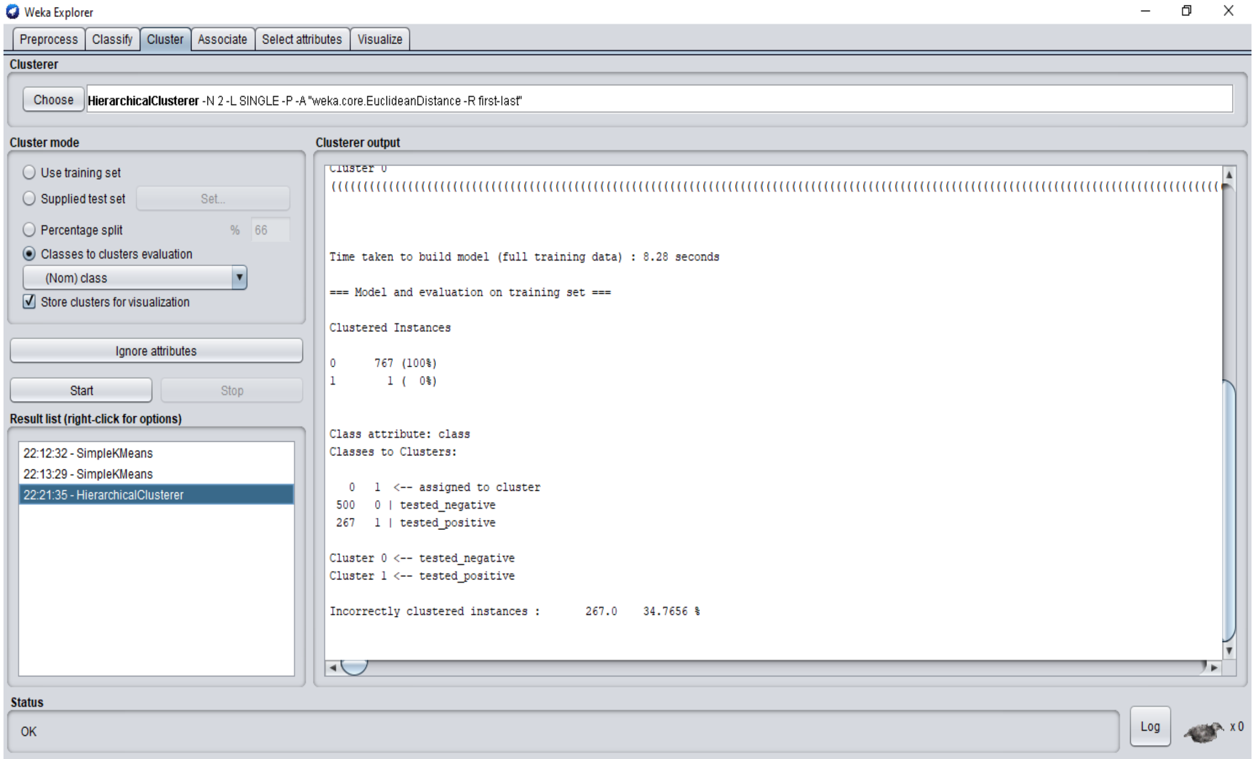
Clusterer output

```
age              33.2409    26.7728    46.4071



Time taken to build model (full training data) : 0.08 seconds

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

Clustered Instances

0      515 ( 67%)
1      253 ( 33%)


Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 380 120 | tested_negative
 135 133 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       255.0    33.2031 %
```

Status

OK

SimpleKmeans



Clusterer: HierarchicalClusterer -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Cluster mode:

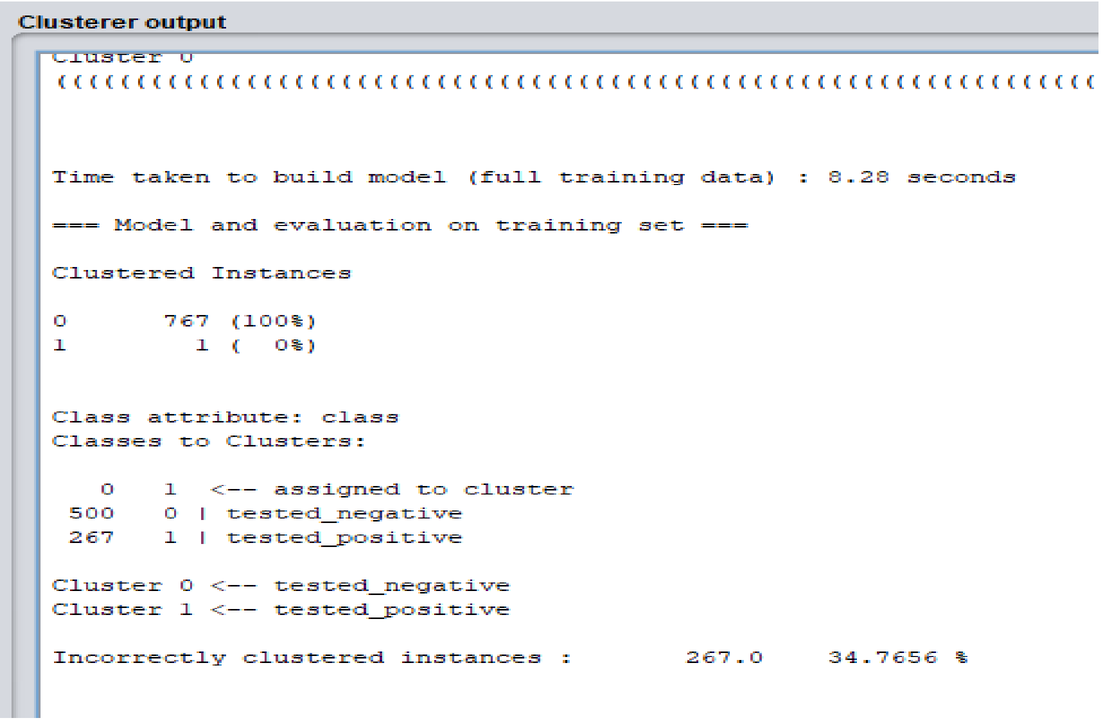
- ○ Use training set
- ○ Supplied test set
- ○ Percentage split % 66
- ● Classes to clusters evaluation
- (Nom) class
- ☑ Store clusters for visualization

Result list (right-click for options):

- 22:12:32 - SimpleKMeans
- 22:13:29 - SimpleKMeans
- 22:21:35 - HierarchicalClusterer

```
Cluster 0
((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((



Time taken to build model (full training data) : 8.28 seconds

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

Clustered Instances

0      767 (100%)
1        1 (  0%)


Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 500   0 | tested_negative
 267   1 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       267.0    34.7656 %
```

Status: OK



```
Cluster 0
(((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((((



Time taken to build model (full training data) : 8.28 seconds

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

Clustered Instances

0      767 (100%)
1        1 (  0%)


Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 500   0 | tested_negative
 267   1 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       267.0    34.7656 %
```

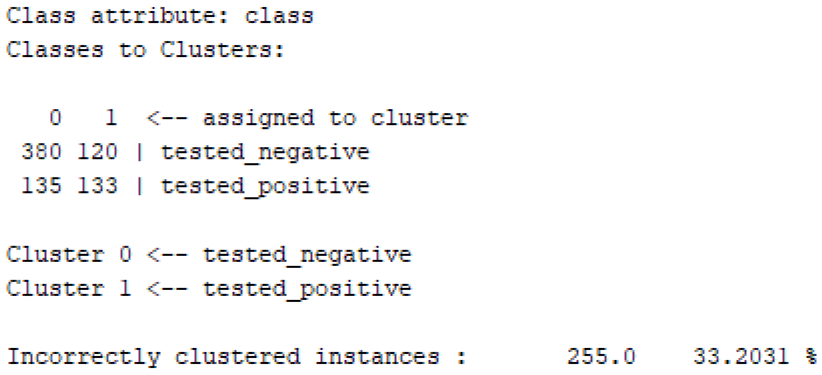
Hierarchical Clustering

In clustering the dataset by **KMeans Clustering**, we classified the dataset into 2 clusters:

Cluster 0 – tested_negative
Cluster 1 – tested_positive

Out of which a total of 515 instances we classified into Cluster0 and 253 instances were classified into Cluster 1. But out of 515 instances in the Cluster 0, 135 instances were wrongly classified as those were labelled tested_positive in the actual dataset. Similarly in the cluster1 120 instances were wrongly classified as those were labelled as tested_negative in the original dataset.

Thus, overall the incorrectly clustered instances were 255 misclassified instances which results to 33.203% error.



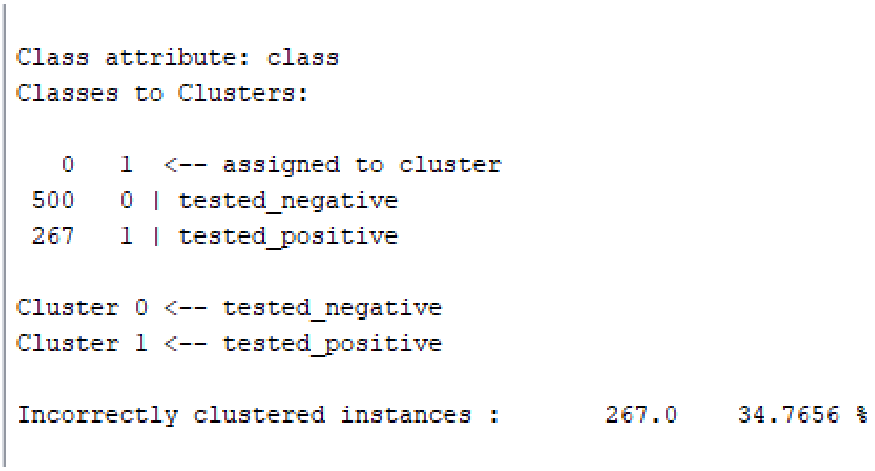
```
Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 380 120 | tested_negative
 135 133 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :	255.0	 33.2031 %
```

In clustering the dataset by **Hierarchical Clustering**, we classified the dataset into 2 clusters:

Cluster 0 – tested_negative
Cluster 1 – tested_positive

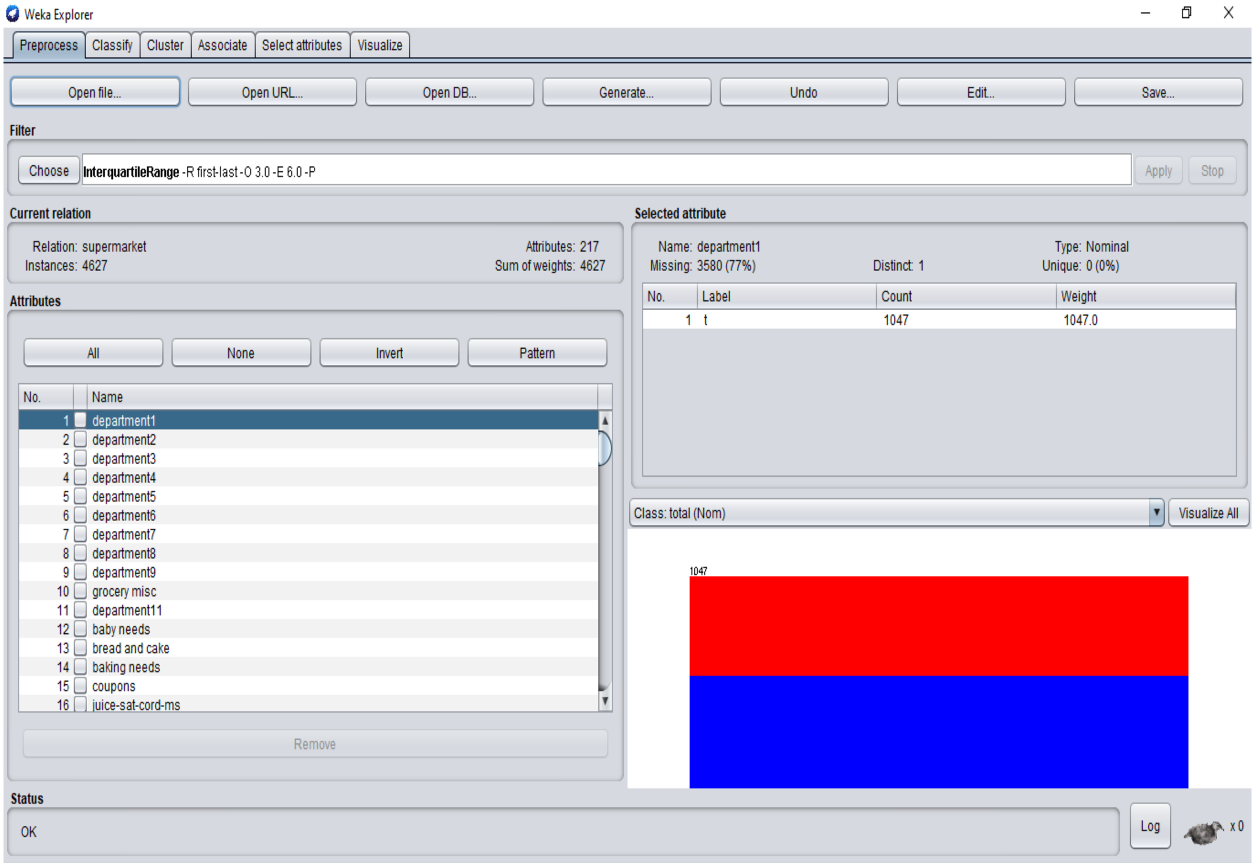
Out of which a total of 767 instances we classified into Cluster0 and 1 data instance was classified into Cluster 1. But out of 767 instances in the Cluster 0, 267 instances were wrongly classified as those were labelled tested_positive in the actual dataset. In the cluster1, 0 instances were wrongly classified.

Thus, overall the incorrectly clustered instances were 255 misclassified instances which results to 34.7656% error.



```
Class attribute: class
Classes to Clusters:

   0   1  <-- assigned to cluster
 500   0 | tested_negative
 267   1 | tested_positive

Cluster 0 <-- tested_negative
Cluster 1 <-- tested_positive

Incorrectly clustered instances :       267.0    34.7656 %
```

6. **Association rule mining:** Perform apriori algo and show the rules created



Filter

InterquartileRange -R first-last -O 3.0 -E 6.0 -P

Current relation

Relation: supermarket
Instances: 4627
Attributes: 217
Sum of weights: 4627

Selected attribute

Name: department1
Missing: 3580 (77%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)

No.	Label	Count	Weight
1	t	1047	1047.0

Attributes

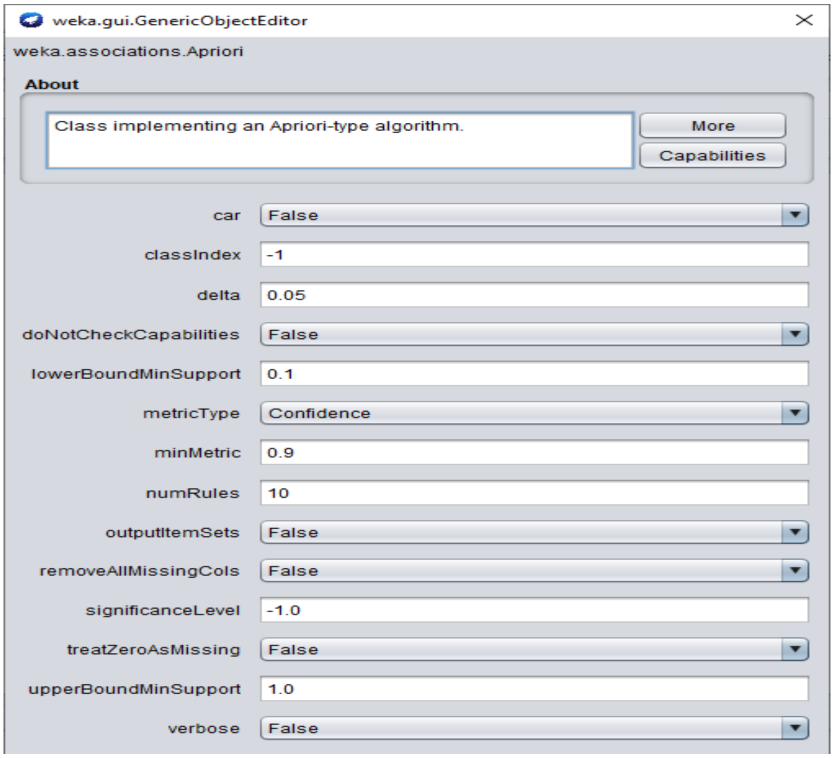
No.	Name
1	department1
2	department2
3	department3
4	department4
5	department5
6	department6
7	department7
8	department8
9	department9
10	grocery misc
11	department11
12	baby needs
13	bread and cake
14	baking needs
15	coupons
16	juice-sat-cord-ms

Class: total (Nom)

Status

OK

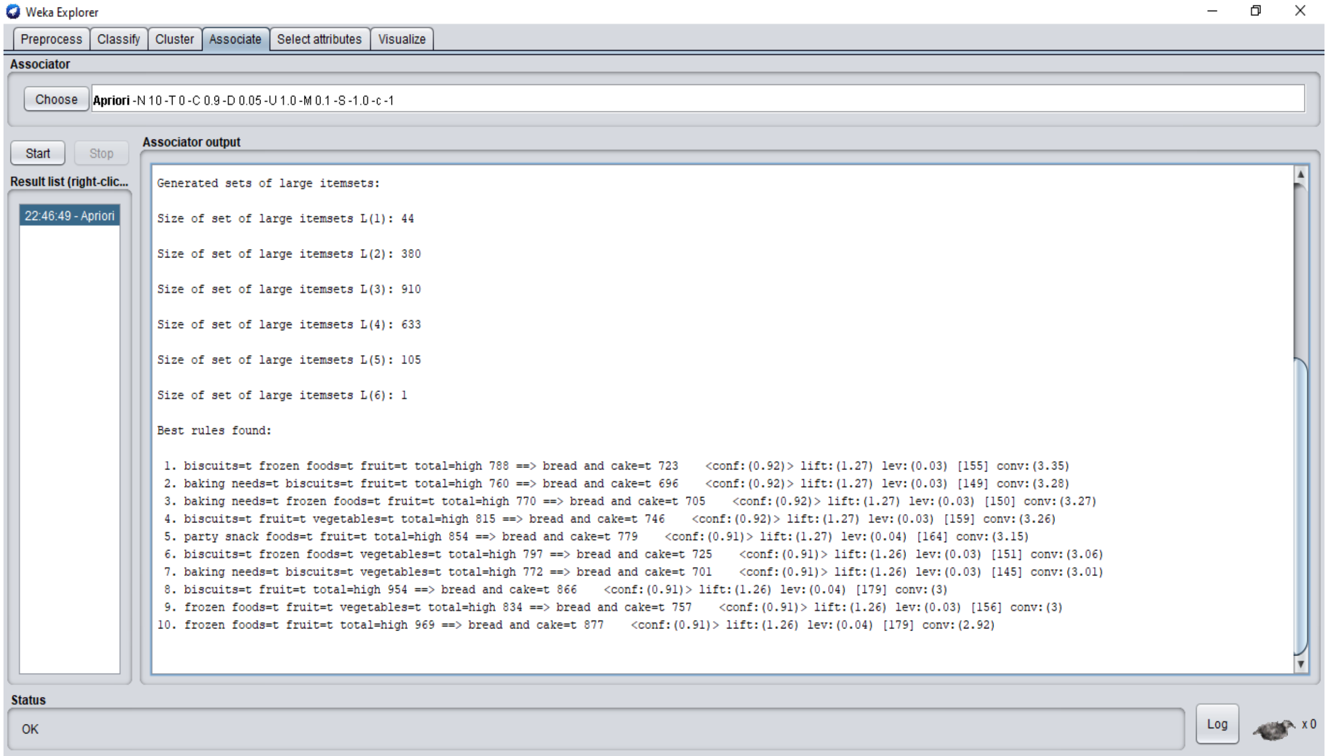
Supermarket dataset



weka.associations.Apriori

Class implementing an Apriori-type algorithm.

car	False
classIndex	-1
delta	0.05
doNotCheckCapabilities	False
lowerBoundMinSupport	0.1
metricType	Confidence
minMetric	0.9
numRules	10
outputItemSets	False
removeAllMissingCols	False
significanceLevel	-1.0
treatZeroAsMissing	False
upperBoundMinSupport	1.0
verbose	False



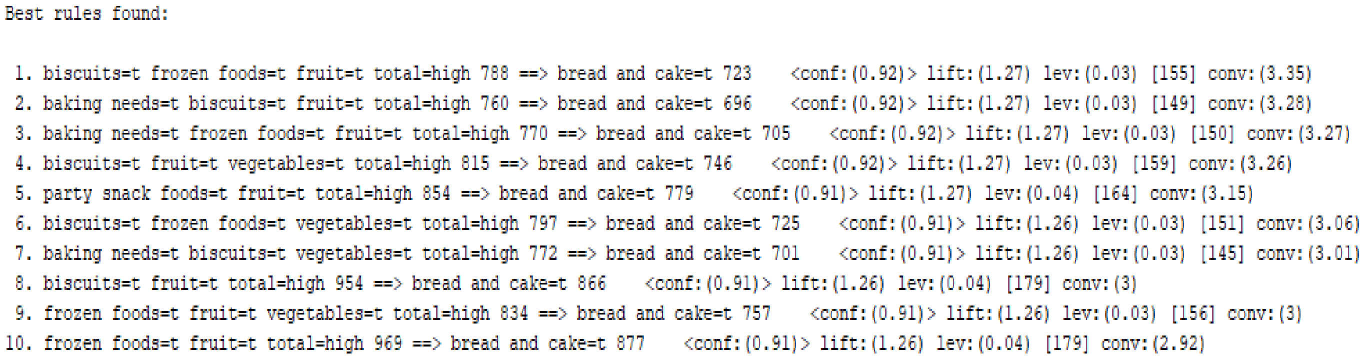
Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

22:46:49 - Apriori

```
Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)
```

OK

Association Rules generated by Apriori Analysis



```
Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)
```

Conclusion:

In this experiment we learnt to about Weka tool used for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools. Using data visualization we can see the correlation between the features and also the distribution of data for each feature thereby, performing necessary pre-processing needed, like transformation to normalize the skewness in the data. Other pre-processing being data cleaning like removing outliers (i.e those values outside the 1.5IQR range). Also, through Select Attribute we got the preference of each attribute with respect to InfoGain. Then I performed Naives Bayes and Random Forest Classification on the dataset using cross validation. I also performed Clustering Algorithms like Simple KMeans and Hierarchical Clustering and observed the numbers of data instances were correctly and wrongly classified into the new clusters. Finally, performed Association mining using Apriori Algorithm and displayed the association rules.