

LAB – 02 - SPOT

Change atleast one parameter in each of the algorithms illustrated as part of lab exercise. Show the change in performance.

Dataset:-

[Index of /ml/machine-learning-databases/tic-tac-toe \(uci.edu\)](http://index.of/ml/machine-learning-databases/tic-tac-toe(uci.edu))

Algorithms:-**1. AdaBoostM1**

With default attributes:-

Batch size-100, num of iterations: 10, classifier: Decision stump

The screenshot shows the Weka Explorer interface with the AdaBoostM1 classifier selected. The 'Test options' section shows 'Cross-validation' with 'Folds' set to 10. The 'Classifier output' section displays the following results:

```

om is missing
positive      negative
0.4936327107693569  0.5063672892306431

Weight: 0.21
Number of performed Iterations: 10
Time taken to build model: 0.09 seconds

=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      695      72.547 %
Incorrectly Classified Instances    263      27.453 %
Kappa statistic                    0.3432
Mean absolute error                 0.3381
Root mean squared error            0.4166
Relative absolute error            74.6393 %
Root relative squared error        87.5379 %
Total Number of Instances          958

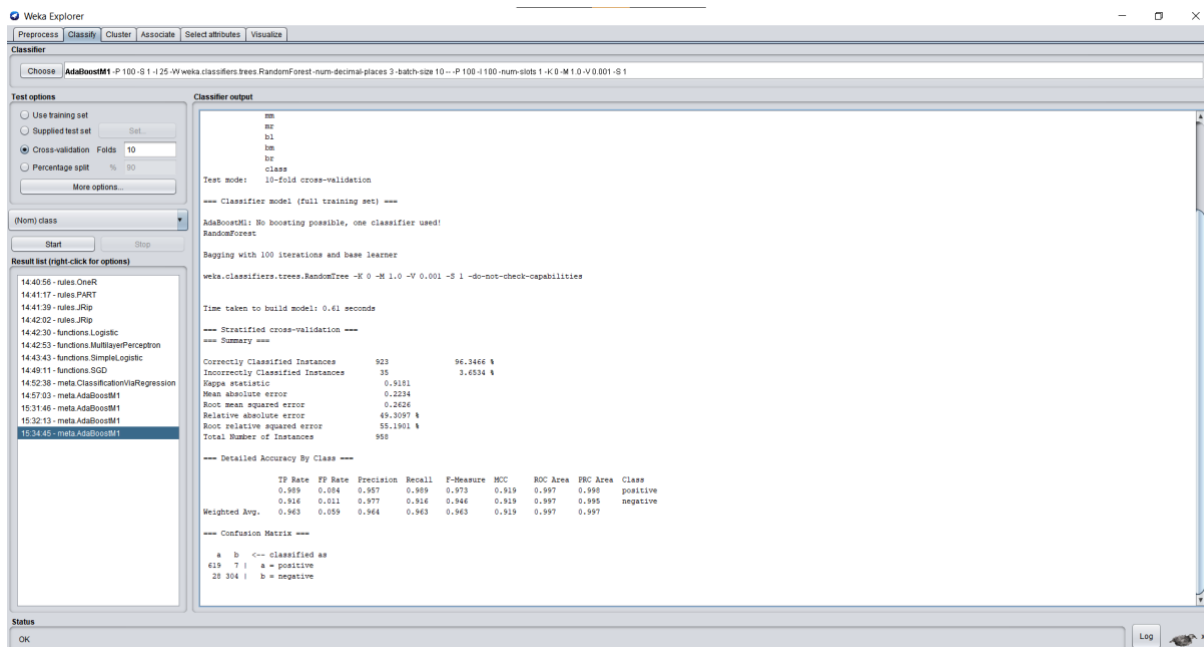
=== Detailed Accuracy By Class ===
               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.877    0.560    0.747     0.877    0.807     0.357    0.794    0.893    positive
               0.440    0.123    0.455     0.440    0.526     0.357    0.794    0.679    negative
Weighted Avg.   0.725    0.409    0.715     0.725    0.710     0.357    0.794    0.819

=== Confusion Matrix ===
  a  b  <-- classified as
549  77 | a = positive
186 146 | b = negative
  
```

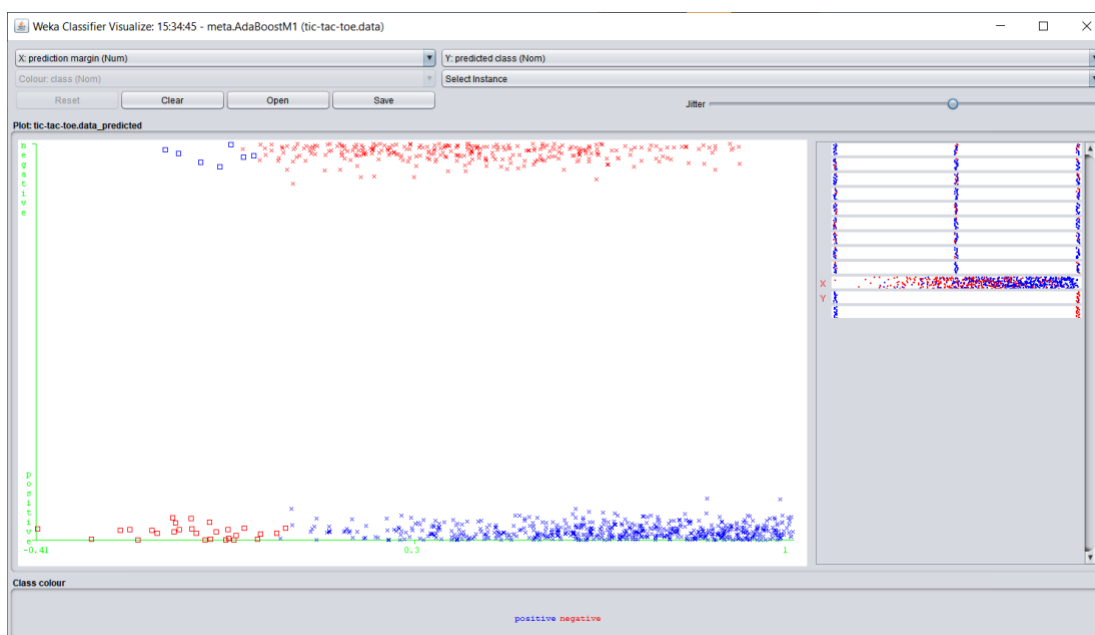
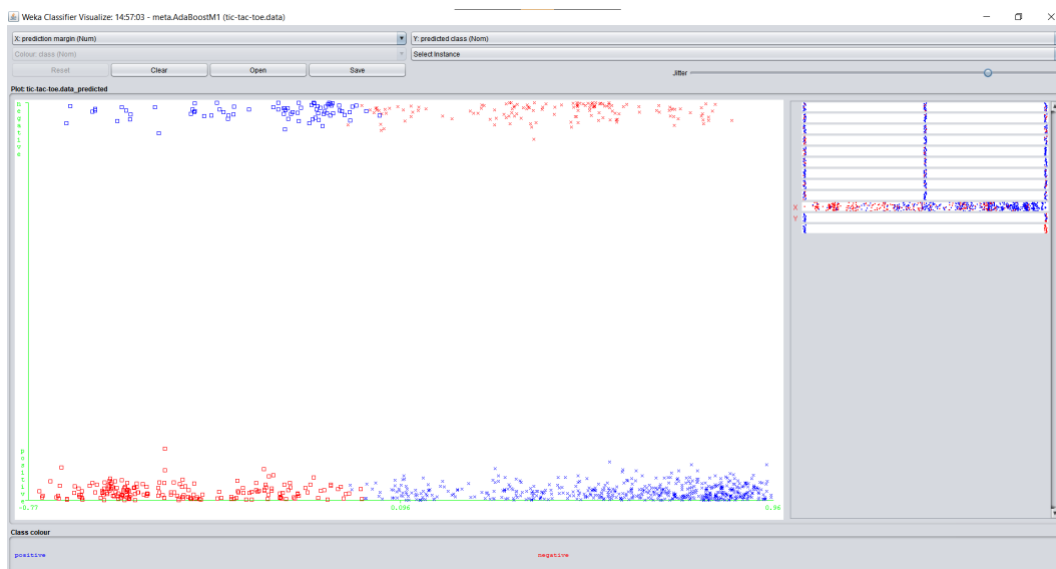
The 'Result list (right-click for options)' on the left shows a list of classifiers, with '14:57:03 - meta AdaBoostM1' selected.

After changing parameters:

Batch size-10, number of iterations: 25, classifier: Random Forest

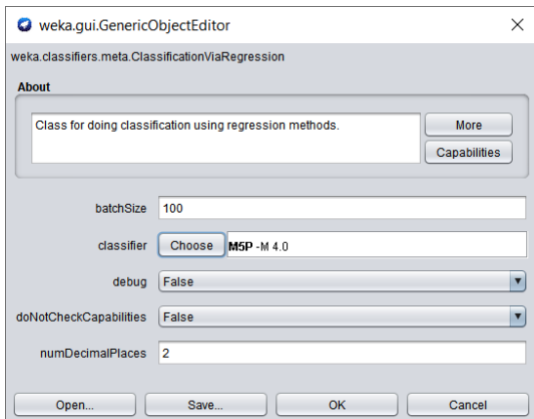


We can see that the confusion matrix has far less deviation and the error has reduced significantly.

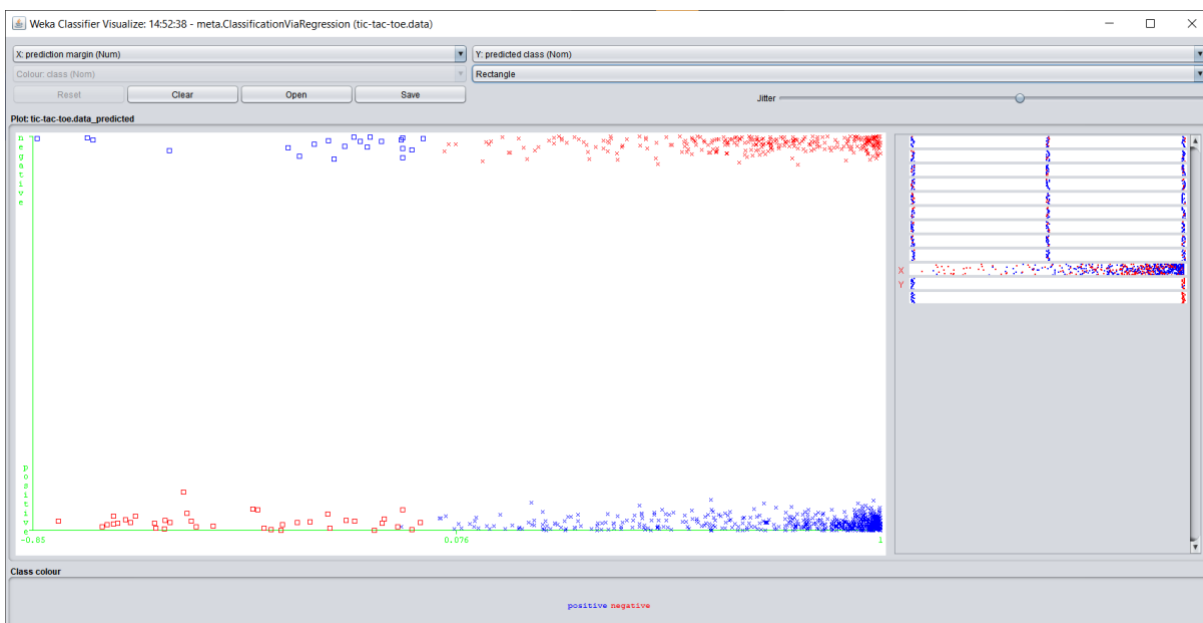
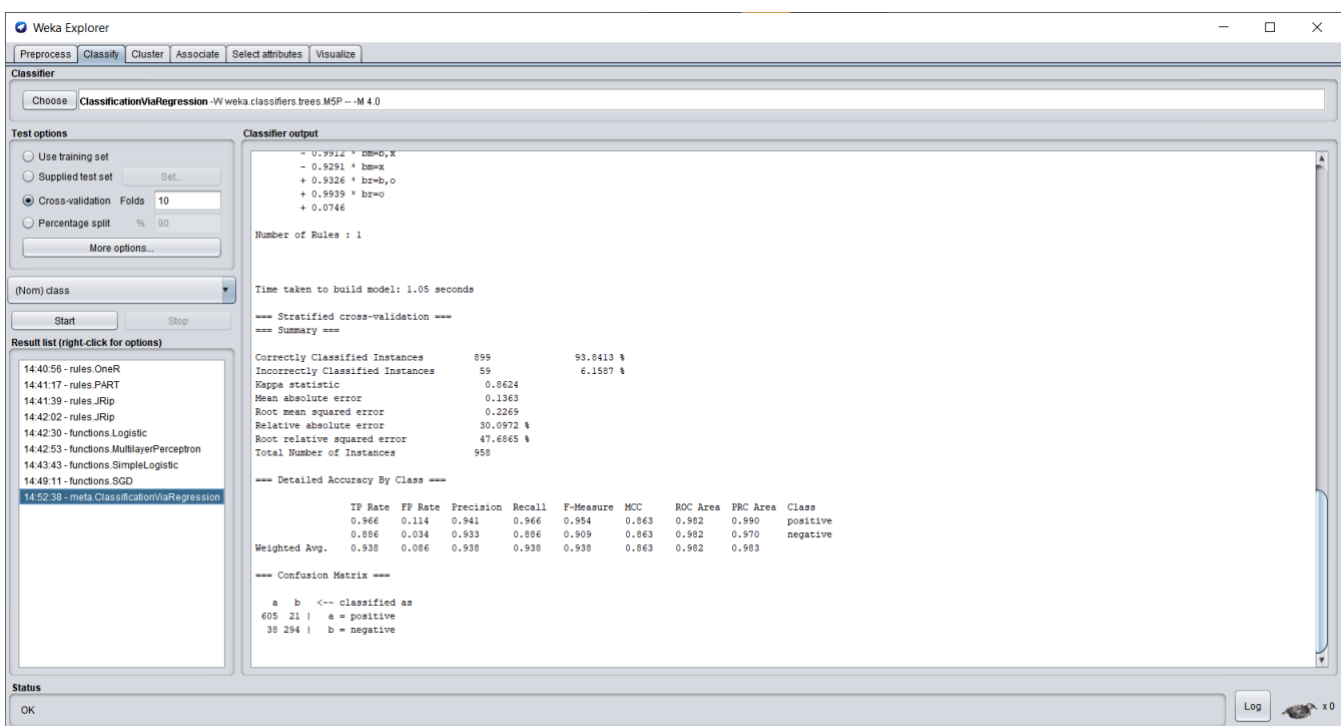


2. Classification via Regression

Using default attributes:-



Performance metrics:-



Changing parameters to:-

Weka Explorer

Preprocess Classify Cluster Associate Select attributes Visualize

Classifier

Choose ClassificationViaRegression -Weka.classifiers.trees.DecisionStump -batch-size 30

Test options

☐ Use training set

☐ Supplied test set

☒ Cross-validation Folds 10

☐ Percentage split % 90

More options...

(Nom) class

Start Stop

Result list (right-click for options)

- 14:40:56 - rules OneR
- 14:41:17 - rules PART
- 14:41:39 - rules JRip
- 14:42:02 - rules JRip
- 14:42:30 - functions Logistic
- 14:42:53 - functions MultiLayerPerceptron
- 14:43:43 - functions SimpleLogistic
- 14:49:11 - functions SGD
- 14:52:38 - meta ClassificationViaRegression
- 14:57:03 - meta AdaBoostM1
- 15:31:48 - meta AdaBoostM1
- 15:32:13 - meta AdaBoostM1
- 15:34:45 - meta AdaBoostM1
- 15:40:45 - functions Logistic
- 15:42:43 - meta ClassificationViaRegression

Classifier output

Classifier for class with index 1:

Decision Stump

Classifications

```

== o : 0.43529411764705893
== i= o : 0.7734627831715211
== is missing : 0.6534446764091058

```

Time taken to build model: 0.02 seconds

=== Stratified cross-validation ===

=== Summary ===

Correctly Classified Instances	670	69.9374 %
Incorrectly Classified Instances	288	30.0626 %
Kappa statistic	0.34	
Mean absolute error	0.4007	
Root mean squared error	0.4477	
Relative absolute error	89.4417 %	
Root relative squared error	94.0739 %	
Total Number of Instances	958	

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Weighted Avg.	0.764	0.422	0.773	0.764	0.768	0.340	0.657	0.744	positive
	0.578	0.236	0.565	0.578	0.571	0.340	0.657	0.481	negative

=== Confusion Matrix ===

```

a b <-- classified as
478 148 | a = positive
140 192 | b = negative

```

weka.gui.GenericObjectEditor

weka.classifiers.meta.ClassificationViaRegression

About

Class for doing classification using regression methods.

More

Capabilities

batchSize 30

classifier Choose DecisionStump

debug False

doNotCheckCapabilities False

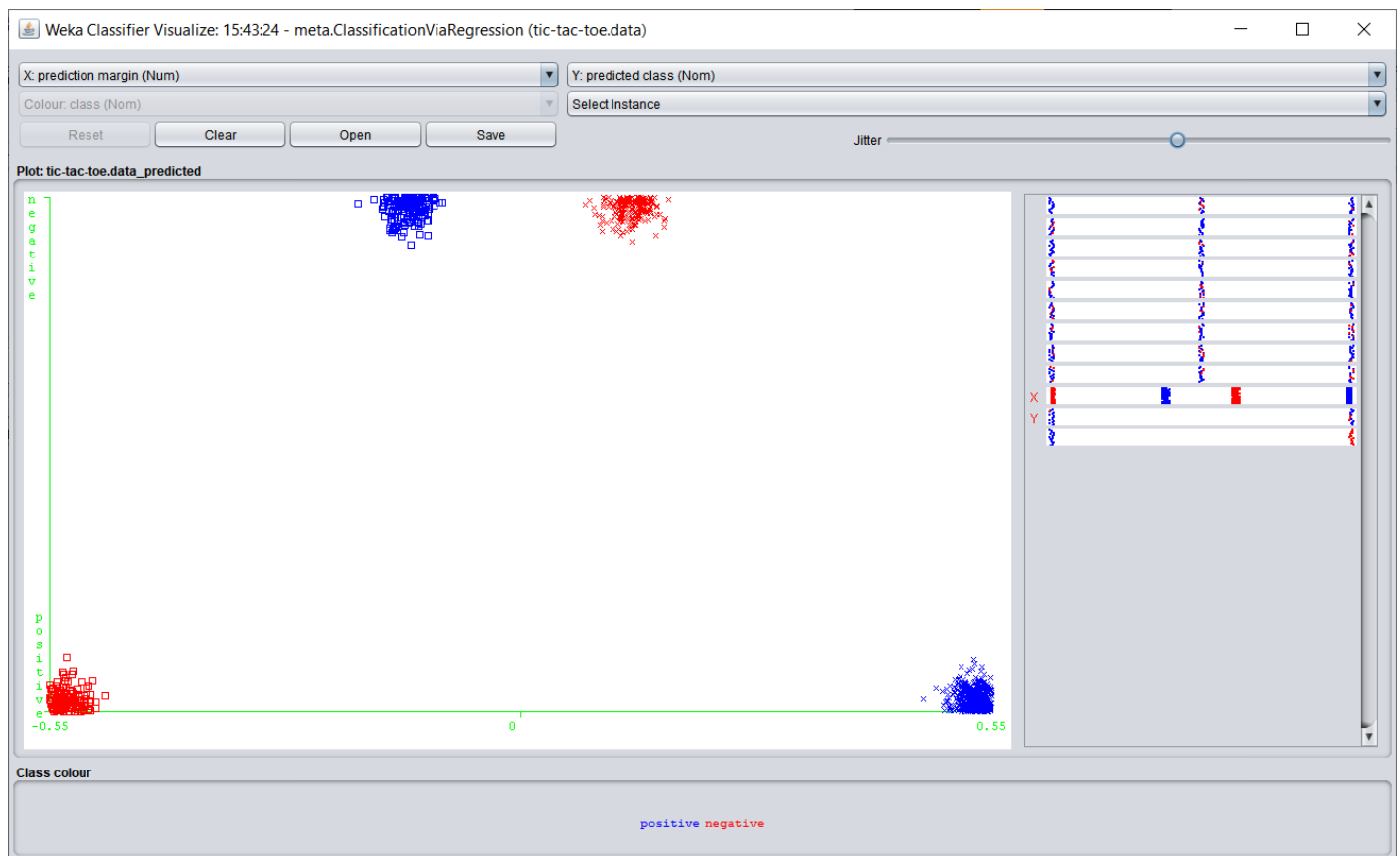
numDecimalPlaces 2

Open... Save... OK Cancel

Status

OK

Log



Changed classifier from M5P to Decision stump has decreased the accuracy of the model significantly. But the values are not spread out in the prediction margin vs class graph.

Clustering:-

1. Simple K-Means

Default Attributes:-

Clusterer

Choose: **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ☐ Use training set
- ☐ Supplied test set
- ☐ Percentage split
- ☒ Classes to clusters evaluation

(Nom) class

☒ Store clusters for visualization

Clusterer output

Cluster 0: x,x,x,b,o,b,o,o,x
Cluster 1: x,x,x,o,b,x,b,o,o

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data	Cluster#	0	1
	(958.0)		(540.0)	(418.0)
t1	x	x	o	
tm	x	x	x	x
tr	x	x	x	x
ml	x	x	o	
mm	x	o	x	
mr	x	o	x	
bl	x	o	x	x
bm	x	o	x	x
br	x	x	o	

Time taken to build model (full training data) : 0.12 seconds

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

Clustered Instances

	0	540 (56%)
1	418 (44%)	

Class attribute: class

Classes to Clusters:

	0	1	<-- assigned to cluster
341	285	1	positive
199	133	1	negative

Cluster 0 <-- negative
Cluster 1 <-- positive

Incorrectly clustered instances : 474.0 49.4761 %

weka.gui.GenericObjectEditor

weka clusters.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-l**

doNotCheckCapabilities: False

doNotReplaceMissingValues: False

fastDistanceCalc: False

initializationMethod: Random

maxIterations: 500

numClusters: 2

numExecutionSlots: 1

preserveInstancesOrder: False

reduceNumberOfDistanceCalcsViaCanopies: False

seed: 10

Open... Save... OK Cancel

After changing attributes:-

Here I am forcing it to cluster into 3 clusters and distance function has been changed.

Clusterer

Choose: **SimpleKMeans** -init 3 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.ManhattanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode

- ☐ Use training set
- ☐ Supplied test set
- ☐ Percentage split
- ☒ Classes to clusters evaluation

(Nom) class

☒ Store clusters for visualization

Clusterer output

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data	Cluster#	0	1	2
	(958.0)		(454.0)	(322.0)	(182.0)
t1	x	x	o	x	
tm	x	x	o	b	
tr	x	o	x	x	
ml	x	o	x	b	
mm	x	x	o	x	
mr	x	o	x	b	
bl	x	o	x	x	
bm	x	x	b	o	
br	x	x	x	o	

Time taken to build model (full training data) : 0.09 seconds

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

Clustered Instances

	0	454 (47%)
1	322 (34%)	
2	182 (19%)	

Class attribute: class

Classes to Clusters:

	0	1	2	<-- assigned to cluster
287	196	143	1	positive
167	126	39	1	negative

Cluster 0 <-- positive
Cluster 1 <-- negative
Cluster 2 <-- No class

Incorrectly clustered instances : 545.0 56.8594 %

weka.gui.GenericObjectEditor

weka clusters.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: **ManhattanDistance -R first**

doNotCheckCapabilities: False

doNotReplaceMissingValues: False

fastDistanceCalc: False

initializationMethod: Farthest first

maxIterations: 500

numClusters: 3

numExecutionSlots: 1

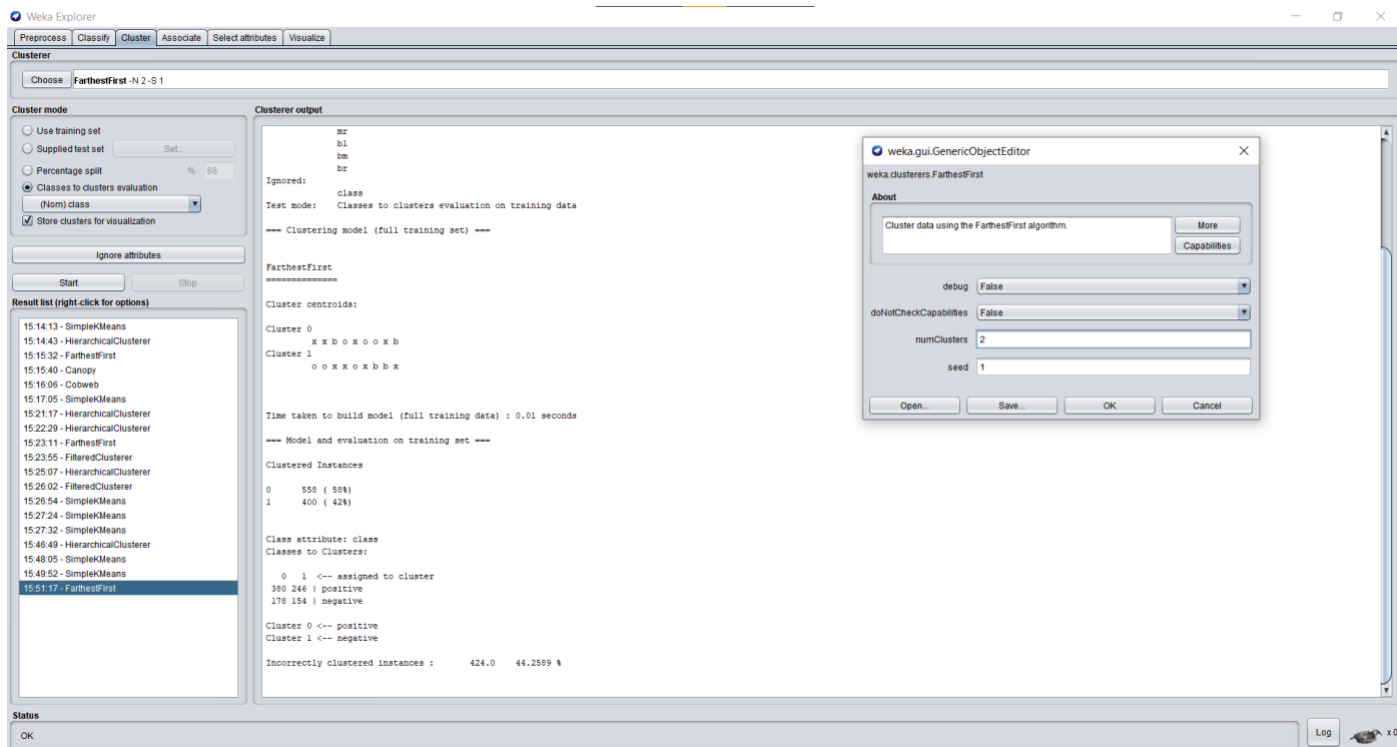
preserveInstancesOrder: False

reduceNumberOfDistanceCalcsViaCanopies: False

seed: 10

Open... Save... OK Cancel

2. Farthest First



After changing attributes:-

On increasing number of clusters, the values are become increasingly random.

