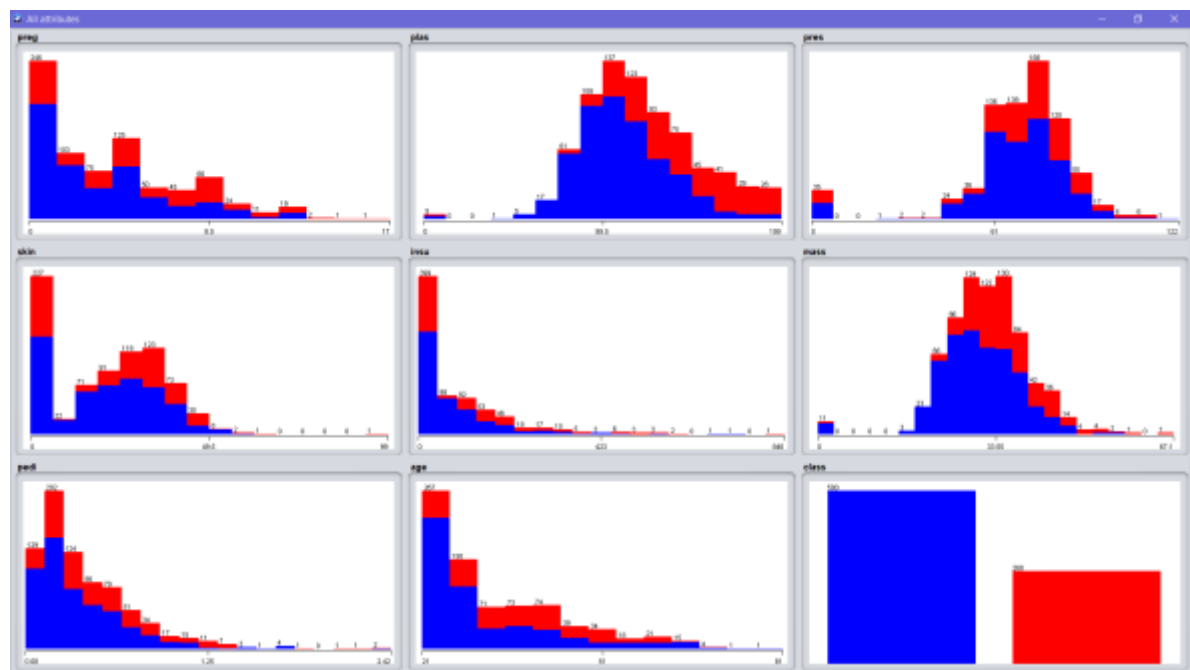
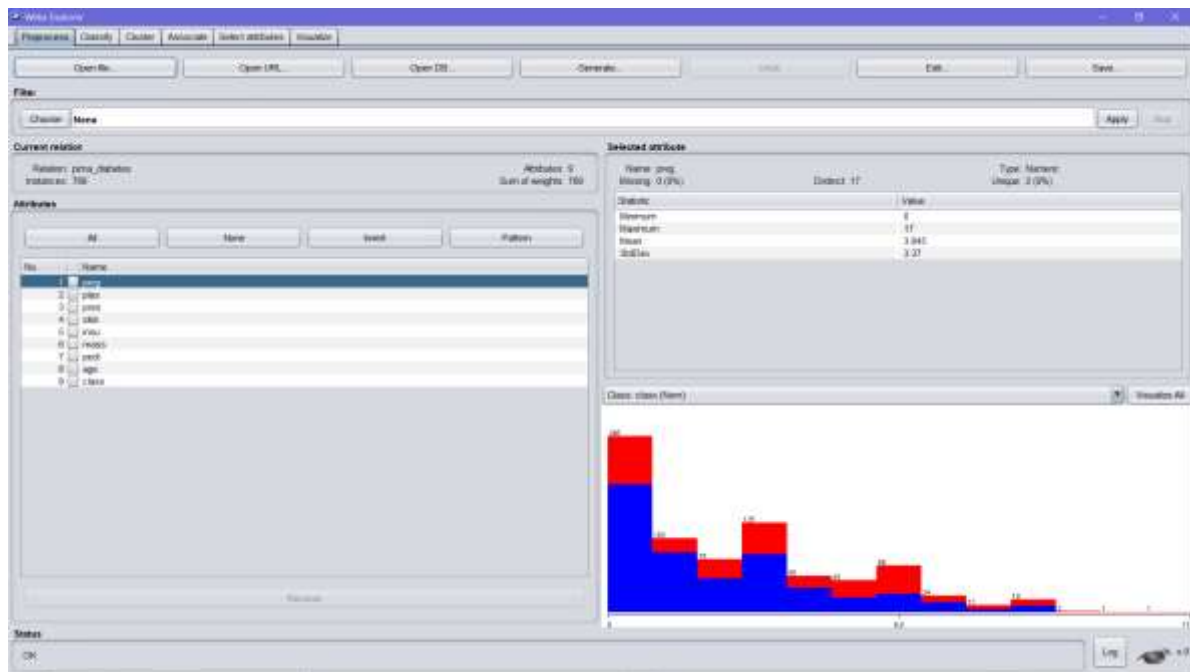


Name: Mareena Fernandes

TE IT	Roll number : 8669
Expt. number : 6	Date of implementation: 10/05/2021
Aim : To analyze and evaluate the performance of different clustering algorithms using WEKA (data mining tool)	
Related Course outcome : CO3 Upon completion of this course students will be able to evaluate the performance of different data mining algorithms using latest tools	
<p>Theory : WEKA contains "clusterers" for finding groups of similar instances in a dataset. The clustering schemes available in WEKA are k-means, Cobwebs, DBSCAN, OPTICS. Clusters can be visualized and compared to true clusters. Evaluation is based on log likelihood if clustering scheme produces a probability distribution. In 'preprocess' window click on 'open file...' button to select data file.</p> <p>Choosing Clustering scheme : In the 'clusterer' box click on 'choose' button. In pull-down menu select WEKA — Clusteres, and select the cluster scheme 'simple K means'. Some implementations of K -means only allow numerical values for attributes ; therefore we do not need to use a filter.</p> <p>Once the clustering algorithm is chosen, right click on algorithm, 'weak.gui.GenericObjectEditor' comes up to the screen. Set the value in 'numclusters' box to number of clusters required. The seed value is used in generating a random number, which is used for making the initial assignments of instances to clusters. Before we run the clustering algorithm, we need to select 'cluster mode'. Click on 'Classes to cluster evaluation' radio-button in 'Cluster mode' box. Click the start button to run the program. When training set is complete, the 'Cluster' output area on the right panel of 'Cluster' window is filled with text describing the results of training and testing. A new entry appears in the 'Result list' box on the left of the result. Run information gives the information about : the clustering scheme used, the relation name, the number of instances, number of attributes. The clustering model shows the centroid of each cluster and statistics on the number and percentage of instances assigned to different clusters. Cluster centroid is the mean vector of each cluster so each dimension value and centroid represents mean value for that dimension in the cluster. Thus centroids can be used to characterize the cluster.</p> <p>Another way of representation of results of clustering is through visualization. Right click on the entry in the 'Result list' and select ' Visualize cluster assignments' in the pull-down window. This brings up Weka clusterer visualize window. This window displays clusters in different colors for better visibility.</p>	

The other density based clustering methods such as DBSCAN and OPTICS are also analyzed and compared.

Dataset:



1. Clustering K-means

- Use training set

=== 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 500 -num-slots 1 -S 10

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: evaluate on training data

=== Clustering model (full training set) ===

kMeans

=====

Number of iterations: 4

Within cluster sum of squared errors: 149.5177664581119

Initial starting points (random):

Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative

Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Cluster#		
	Full Data (768.0)	0 (500.0)	1 (268.0)
=====			
preg	3.8451	3.298	4.8657
plas	120.8945	109.98	141.2575
pres	69.1055	68.184	70.8246
skin	20.5365	19.664	22.1642
insu	79.7995	68.792	100.3358
mass	31.9926	30.3042	35.1425
pedi	0.4719	0.4297	0.5505
age	33.2409	31.19	37.0672
class	tested_negative	tested_negative	tested_positive

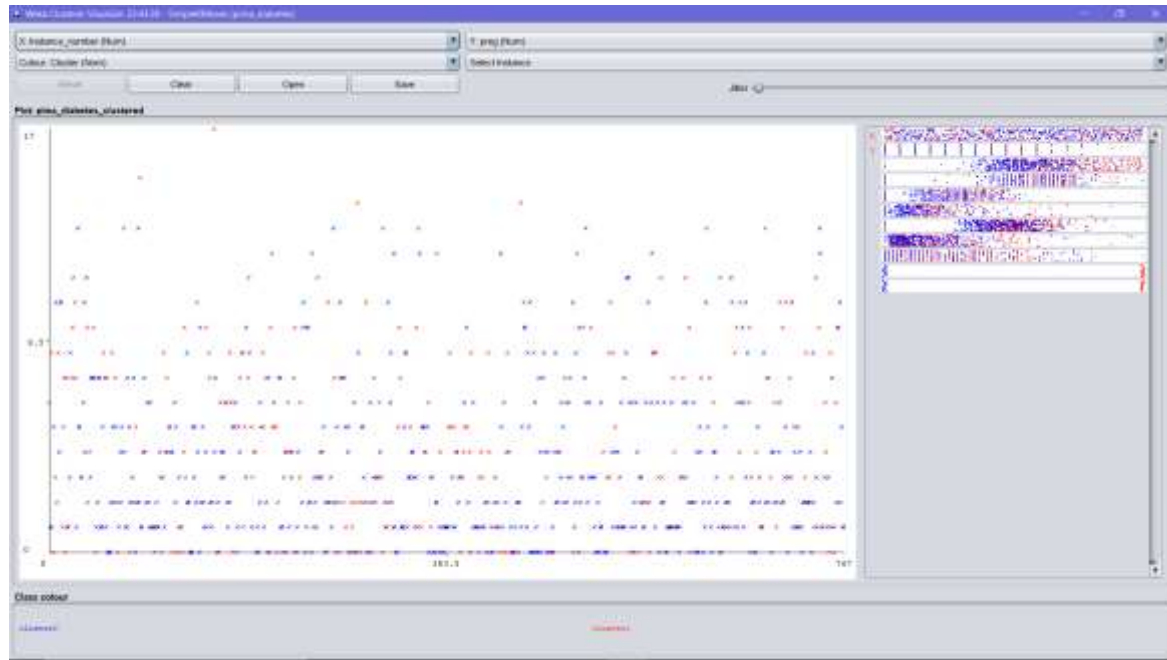
Time taken to build model (full training data) : 0.03 seconds

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

Clustered Instances

0 500 (65%)

1 268 (35%)



- Percentage split

=== 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 500 -num-slots 1 -S 10

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: split 66% train, remainder test

=== Clustering model (full training set) ===

kMeans

=====

Number of iterations: 4

Within cluster sum of squared errors: 149.5177664581119

Initial starting points (random):

Cluster 0: 1,126,56,29,152,28.7,0.801,21,tested_negative

Cluster 1: 8,95,72,0,0,36.8,0.485,57,tested_negative

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Cluster#		
	Full Data (768.0)	0 (500.0)	1 (268.0)
=====			
preg	3.8451	3.298	4.8657
plas	120.8945	109.98	141.2575
pres	69.1055	68.184	70.8246
skin	20.5365	19.664	22.1642
insu	79.7995	68.792	100.3358
mass	31.9926	30.3042	35.1425
pedi	0.4719	0.4297	0.5505
age	33.2409	31.19	37.0672
class	tested_negative	tested_negative	tested_positive

Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on test split ===

kMeans

=====

Number of iterations: 2

Within cluster sum of squared errors: 115.21146310581545

Initial starting points (random):

Cluster 0: 10,148,84,48,237,37.6,1.001,51,tested_positive

Cluster 1: 6,154,74,32,193,29.3,0.839,39,tested_negative

Missing values globally replaced with mean/mode

Final cluster centroids:

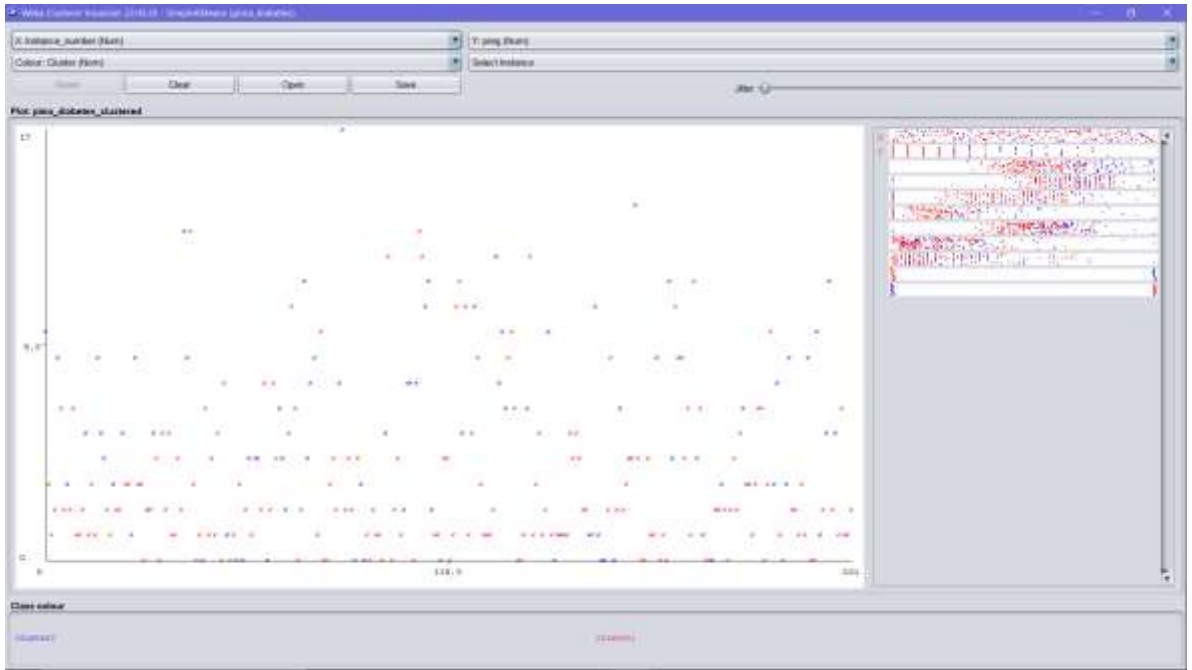
Attribute	Cluster#		
	Full Data (506.0)	0 (184.0)	1 (322.0)
=====			
preg	3.919	4.7826	3.4255
plas	121.164	139.3967	110.7453
pres	69.1779	70.8587	68.2174
skin	19.9486	21.962	18.7981
insu	78.585	90.0217	72.0497
mass	32.0275	35.2766	30.1708
pedi	0.4798	0.5516	0.4388
age	33.7925	37.4674	31.6925
class	tested_negative	tested_positive	tested_negative

Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0 84 (32%)

1 178 (68%)



- Classes to clusters evaluation

=== 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 500 -num-slots 1 -S 10

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

Ignored:

class

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

kMeans

=====

Number of iterations: 7

Within cluster sum of squared errors: 121.2579017999101

Initial starting points (random):

Cluster 0: 1,126,56,29,152,28.7,0.801,21

Cluster 1: 8,95,72,0,0,36.8,0.485,57

Missing values globally replaced with mean/mode

Final cluster centroids:

Cluster#

Attribute	Full Data	0	1
	(768.0)	(515.0)	(253.0)

=====

preg	3.8451	2.0835	7.4308
plas	120.8945	115.3282	132.2253
pres	69.1055	65.9903	75.4466
skin	20.5365	21.8194	17.9249
insu	79.7995	85.0194	69.1739
mass	31.9926	31.7751	32.4352
pedi	0.4719	0.4708	0.4741
age	33.2409	26.7728	46.4071

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

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

Clustered Instances

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

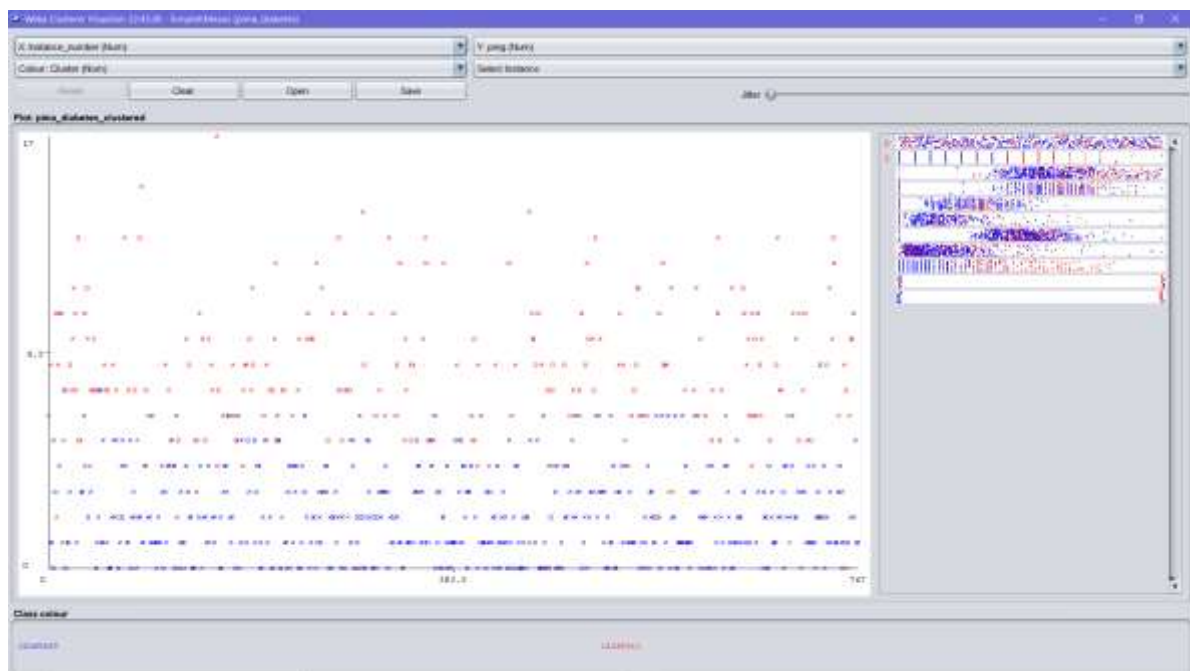
Class attribute: class

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 %



2. DB Scan

- Use training set

=== Run information ===

Scheme: weka.clusterers.DBSCAN -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last"

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: evaluate on training data

=== Clustering model (full training set) ===

DBSCAN clustering results

=====

Clustered DataObjects: 768

Number of attributes: 9

Epsilon: 0.9; minPoints: 6

Distance-type:

Number of generated clusters: 2

Elapsed time: .14

(0.) 6,148,72,35,0,33.6,0.627,50,tested_positive	--> 0
(1.) 1,85,66,29,0,26.6,0.351,31,tested_negative	--> 1
(2.) 8,183,64,0,0,23.3,0.672,32,tested_positive	--> 0
(3.) 1,89,66,23,94,28.1,0.167,21,tested_negative	--> 1
(4.) 0,137,40,35,168,43.1,2.288,33,tested_positive	--> 0
(5.) 5,116,74,0,0,25.6,0.201,30,tested_negative	--> 1
(6.) 3,78,50,32,88,31,0.248,26,tested_positive	--> 0
.	
.	
.	
(761.) 9,170,74,31,0,44,0.403,43,tested_positive	--> 0
(762.) 9,89,62,0,0,22.5,0.142,33,tested_negative	--> 1
(763.) 10,101,76,48,180,32.9,0.171,63,tested_negative	--> 1
(764.) 2,122,70,27,0,36.8,0.34,27,tested_negative	--> 1
(765.) 5,121,72,23,112,26.2,0.245,30,tested_negative	--> 1
(766.) 1,126,60,0,0,30.1,0.349,47,tested_positive	--> 0
(767.) 1,93,70,31,0,30.4,0.315,23,tested_negative	--> 1

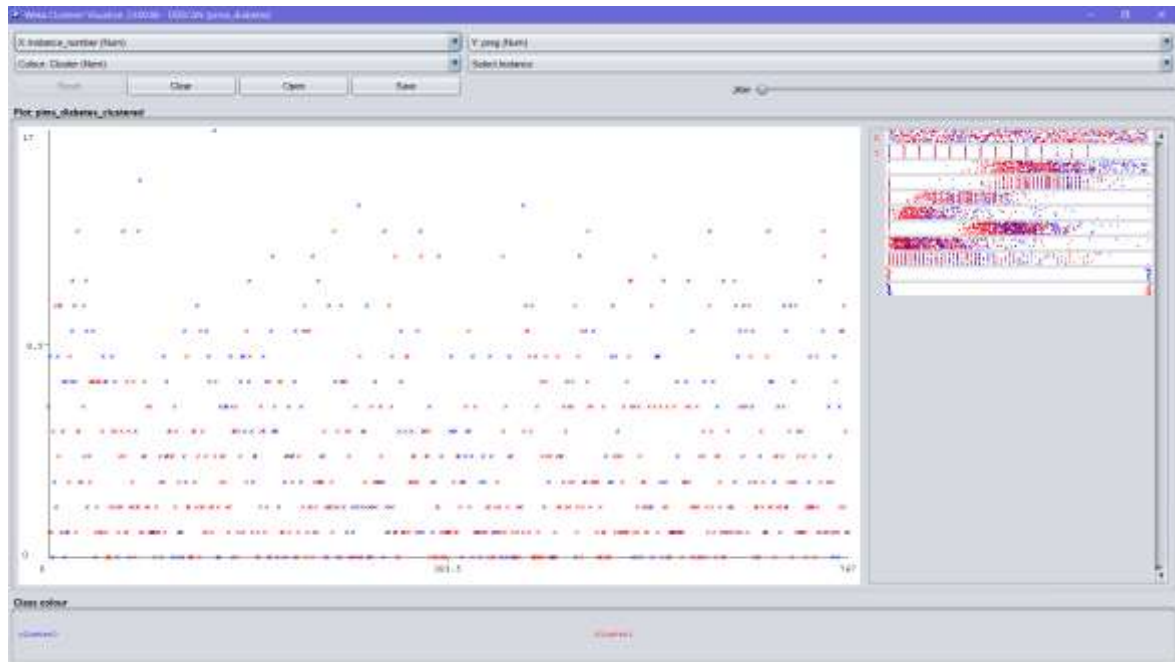
Time taken to build model (full training data) : 0.14 seconds

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

Clustered Instances

0 268 (35%)

1 500 (65%)



- Percentage split

=== Run information ===

Scheme: weka.clusterers.DBSCAN -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last"

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: split 66% train, remainder test

=== Clustering model (full training set) ===

DBSCAN clustering results

=====

Clustered DataObjects: 768

Number of attributes: 9

Epsilon: 0.9; minPoints: 6

Distance-type:

Number of generated clusters: 2

Elapsed time: .09

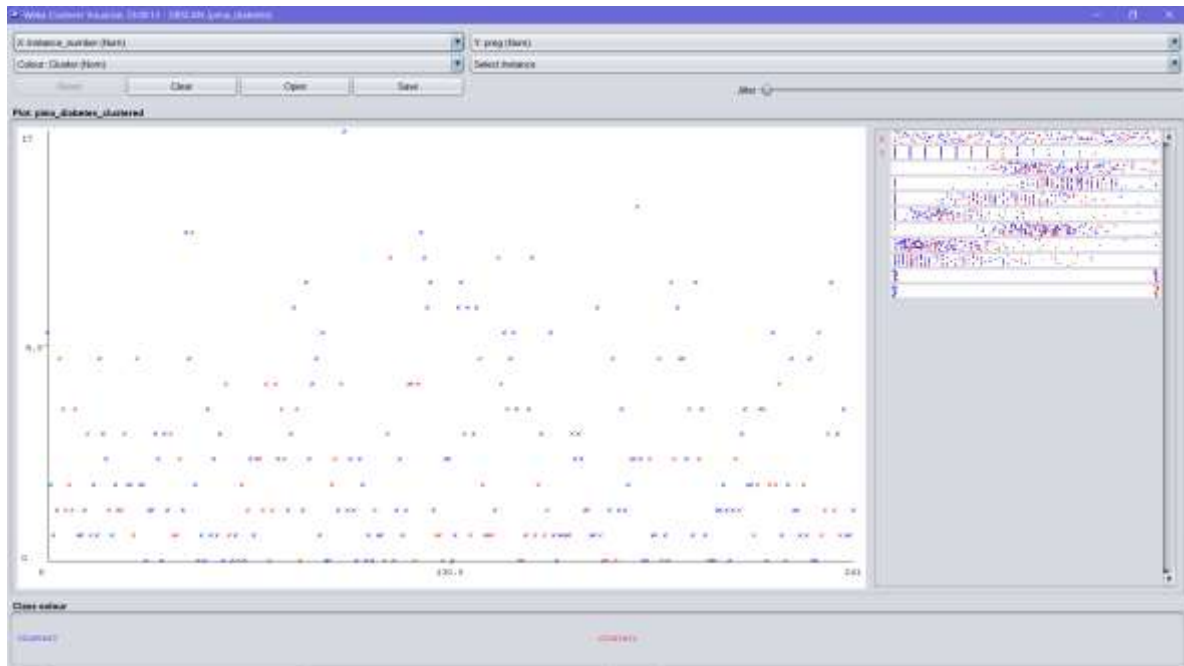
(0.) 6,148,72,35,0,33.6,0.627,50,tested_positive	--> 0
(1.) 1,85,66,29,0,26.6,0.351,31,tested_negative	--> 1
(2.) 8,183,64,0,0,23.3,0.672,32,tested_positive	--> 0
(3.) 1,89,66,23,94,28.1,0.167,21,tested_negative	--> 1
(4.) 0,137,40,35,168,43.1,2.288,33,tested_positive	--> 0
(5.) 5,116,74,0,0,25.6,0.201,30,tested_negative	--> 1
.	
.	
.	
(500.) 1,124,60,32,0,35.8,0.514,21,tested_negative	--> 0
(501.) 2,197,70,99,0,34.7,0.575,62,tested_positive	--> 1
(502.) 0,151,90,46,0,42.1,0.371,21,tested_positive	--> 1
(503.) 7,178,84,0,0,39.9,0.331,41,tested_positive	--> 1
(504.) 2,83,65,28,66,36.8,0.629,24,tested_negative	--> 0
(505.) 3,99,62,19,74,21.8,0.279,26,tested_negative	--> 0

Time taken to build model (percentage split) : 0.07 seconds

Clustered Instances

0 164 (63%)

1 98 (37%)



- Classes to clusters evaluation

=== Run information ===

Scheme: weka.clusterers.DBSCAN -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last"

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

Ignored:

class

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

DBSCAN clustering results

=====

Clustered DataObjects: 768

Number of attributes: 8

Epsilon: 0.9; minPoints: 6

Distance-type:

Number of generated clusters: 1

Elapsed time: .1

(0.)	6,148,72,35,0,33.6,0.627,50	--> 0
(1.)	1,85,66,29,0,26.6,0.351,31	--> 0
(2.)	8,183,64,0,0,23.3,0.672,32	--> 0
(3.)	1,89,66,23,94,28.1,0.167,21	--> 0
(4.)	0,137,40,35,168,43.1,2.288,33	--> 0
(5.)	5,116,74,0,0,25.6,0.201,30	--> 0
.		
.		
.		
(762.)	9,89,62,0,0,22.5,0.142,33	--> 0
(763.)	10,101,76,48,180,32.9,0.171,63	--> 0
(764.)	2,122,70,27,0,36.8,0.34,27	--> 0
(765.)	5,121,72,23,112,26.2,0.245,30	--> 0
(766.)	1,126,60,0,0,30.1,0.349,47	--> 0
(767.)	1,93,70,31,0,30.4,0.315,23	--> 0

Time taken to build model (full training data) : 0.1 seconds

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

Clustered Instances

0 768 (100%)

Class attribute: class

Classes to Clusters:

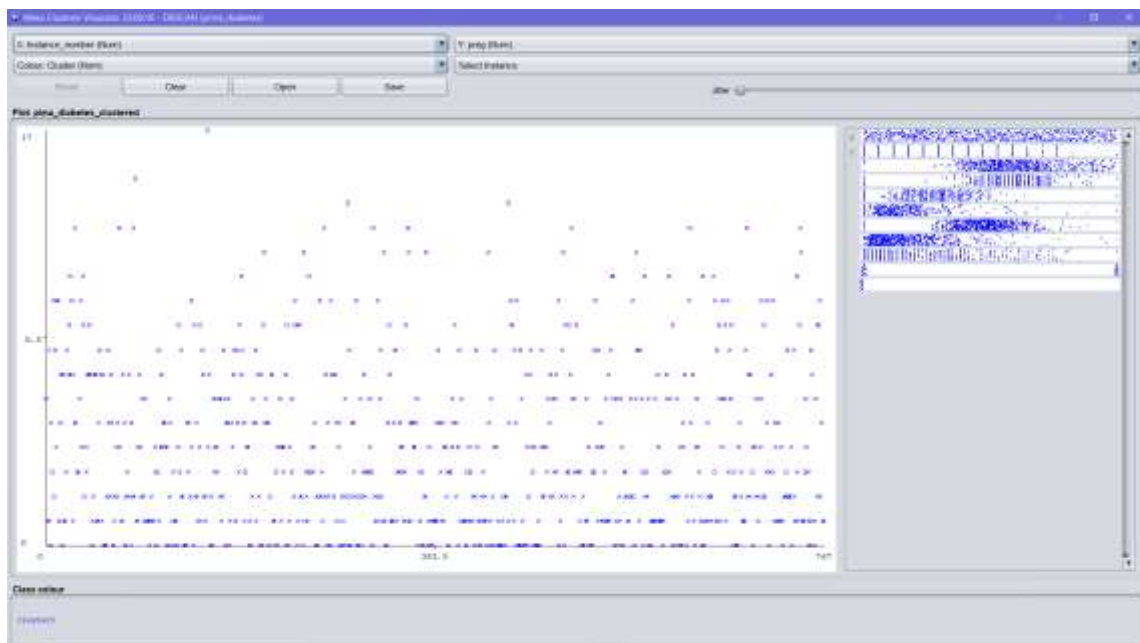
0 <-- assigned to cluster

500 | tested_negative

268 | tested_positive

Cluster 0 <-- tested_negative

Incorrectly clustered instances : 268.0 34.8958 %



3. OPTICS

- Use training set

=== Run information ===

Scheme: weka.clusterers.OPTICS -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last" -db-output .

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: evaluate on training data

=== Clustering model (full training set) ===

OPTICS clustering results

=====

Clustered DataObjects: 768

Number of attributes: 9

Epsilon: 0.9; minPoints: 6

Write results to file: no

Distance-type:

Number of generated clusters: 0

Elapsed time: .19

(0.) 6,148,72,35,0,33.6,0.627,50,tested_posit --> c_dist: 0.284 r_dist:
UNDEFINED
(417.) 4,144,82,32,0,38.5,0.554,37,tested_posit --> c_dist: 0.279 r_dist: 0.284
(171.) 6,134,70,23,130,35.4,0.542,29,tested_pos --> c_dist: 0.257 r_dist: 0.279
(189.) 5,139,80,35,160,31.6,0.361,25,tested_pos --> c_dist: 0.226 r_dist: 0.257
(110.) 3,171,72,33,135,33.3,0.199,24,tested_pos --> c_dist: 0.212 r_dist: 0.226
(132.) 3,170,64,37,225,34.5,0.356,30,tested_pos --> c_dist: 0.212 r_dist: 0.212
(425.) 4,184,78,39,277,37,0.264,31,tested_posit --> c_dist: 0.238 r_dist: 0.212
.
.
.
(622.) 6,183,94,0,0,40.8,1.461,45,tested_negati --> c_dist: 0.535 r_dist: 0.445
(744.) 13,153,88,37,140,40.6,1.174,39,tested_ne --> c_dist: 0.487 r_dist: 0.445
(487.) 0,173,78,32,265,46.5,1.159,58,tested_neg --> c_dist: 0.58 r_dist: 0.447
(58.) 0,146,82,0,0,40.5,1.781,44,tested_negati --> c_dist: 0.545 r_dist: 0.494
(453.) 2,119,0,0,0,19.6,0.832,72,tested_negativ --> c_dist: 0.632 r_dist: 0.499

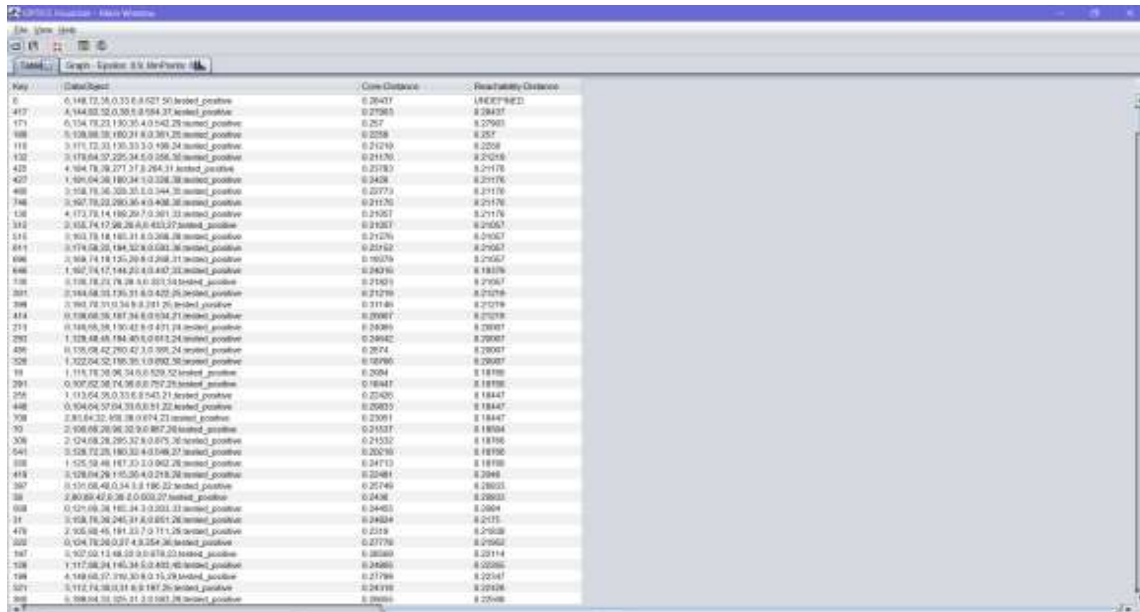
(228.) 4,197,70,39,744,36.7,2.329,31,tested_neg --> c_dist: 0.88 r_dist: 0.776

Time taken to build model (full training data) : 0.39 seconds

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

Clustered Instances

Unclassified instances : 768



Key	DataObject	Core Distance	Reliability Distance
0	0.148,12.18,0.33,0.827,50,tested_positive	0.26437	0.26437
417	0.148,12.18,0.33,0.827,50,tested_positive	0.27985	0.27985
171	0.154,19.23,1.93,35.4,0.543,25,tested_positive	0.257	0.25983
888	0.158,18.18,1.88,21.8,0.361,25,tested_positive	0.258	0.257
118	0.171,12.33,1.35,33.3,0.196,24,tested_positive	0.21269	0.2208
132	0.178,18.37,2.25,34.6,0.156,30,tested_positive	0.21130	0.21268
423	0.184,19.38,2.71,37.8,0.264,31,tested_positive	0.21783	0.21178
427	0.184,19.38,2.71,37.8,0.264,31,tested_positive	0.2438	0.21178
488	0.188,19.46,2.28,35.3,0.144,30,tested_positive	0.20779	0.21178
748	0.197,18.22,2.85,36.9,0.408,38,tested_positive	0.21170	0.21178
138	0.172,17.14,1.88,29.7,0.361,31,tested_positive	0.21527	0.21178
312	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
315	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
811	0.178,18.38,1.84,32.8,0.081,38,tested_positive	0.21527	0.21527
888	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
848	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
138	0.172,17.14,1.88,29.7,0.361,31,tested_positive	0.21527	0.21527
388	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
414	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
213	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
293	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
428	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
528	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
18	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
291	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
258	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
448	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
708	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
793	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
793	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
308	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
541	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
888	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
418	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
387	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
38	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
888	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
31	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
478	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
888	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
187	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
138	0.172,17.14,1.88,29.7,0.361,31,tested_positive	0.21527	0.21527
188	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
527	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527
888	0.188,17.47,2.28,35.3,0.144,30,tested_positive	0.21527	0.21527

- Percentage split

=== Run information ===

Scheme: weka.clusterers.OPTICS -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last" -db-output .

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

class

Test mode: split 66% train, remainder test

=== Clustering model (full training set) ===

OPTICS clustering results

=====

Clustered DataObjects: 768

Number of attributes: 9

Epsilon: 0.9; minPoints: 6

Write results to file: no

Distance-type:

Number of generated clusters: 0

Elapsed time: .14

(0.) 6,148,72,35,0,33.6,0.627,50,tested_posit --> c_dist: 0.284 r_dist:

UNDEFINED

(417.) 4,144,82,32,0,38.5,0.554,37,tested_posit --> c_dist: 0.279 r_dist: 0.284

(171.) 6,134,70,23,130,35.4,0.542,29,tested_pos --> c_dist: 0.257 r_dist: 0.279

(189.) 5,139,80,35,160,31.6,0.361,25,tested_pos --> c_dist: 0.226 r_dist: 0.257

(110.) 3,171,72,33,135,33.3,0.199,24,tested_pos --> c_dist: 0.212 r_dist: 0.226

(132.) 3,170,64,37,225,34.5,0.356,30,tested_pos --> c_dist: 0.212 r_dist: 0.212

(425.) 4,184,78,39,277,37,0.264,31,tested_posit --> c_dist: 0.238 r_dist: 0.212

.

.

.

(74.) 0,137,40,35,168,43.1,2.288,33,tested_pos --> c_dist: 0.649 r_dist: 0.537

(4.) 6,0,68,41,0,39,0.727,41,tested_positive --> c_dist: 0.588 r_dist: 0.549

(175.) 0,180,78,63,14,59.4,2.42,25,tested_posit --> c_dist: 0.752 r_dist: 0.559

(397.) 1,189,60,23,846,30.1,0.398,59,tested_pos --> c_dist: 0.861 r_dist: 0.561

(146.) 13,129,0,30,0,39.9,0.569,44,tested_posit --> c_dist: 0.665 r_dist: 0.565

(469.) 11,135,0,0,0,52.3,0.578,40,tested_positi --> c_dist: 0.658 r_dist: 0.62

(501.) 2,197,70,99,0,34.7,0.575,62,tested_posit --> c_dist: 0.823 r_dist: 0.745

Time taken to build model (percentage split) : 0.29 seconds

Clustered Instances

Unclustered instances : 262

Key	Data Object	Core Outcome	Reactivity Outcome
0	0.148,12.35,0.33,0.021,50,boxed_positive	0.2047	1.90278E21
417	0.144,02.32,0.38,0.054,21,boxed_positive	0.2798	2.29437
174	0.134,12.23,1.92,35.0,0.042,26,boxed_positive	0.257	8.27903
168	0.138,08.38,1.90,21.0,0.381,25,boxed_positive	0.2258	8.257
118	0.171,12.33,1.93,33.0,0.496,24,boxed_positive	0.21249	8.2208
132	0.178,04.37,2.25,34.0,0.056,30,boxed_positive	0.21170	8.21016
432	0.184,19.38,2.77,1.9,294.31,boxed_positive	0.21783	8.21178
427	1.481,04.38,1.90,34.0,0.038,38,boxed_positive	0.2428	8.21176
488	0.158,19.36,3.08,35.0,0.044,30,boxed_positive	0.20773	8.21176
748	0.167,19.22,2.90,36.0,0.408,38,boxed_positive	0.21170	8.21176
138	0.173,19.14,1.90,29.7,0.381,33,boxed_positive	0.21027	8.21176
312	0.152,14.17,2.26,28.0,0.413,37,boxed_positive	0.21037	8.21057
115	0.163,19.18,1.81,31.0,0.038,38,boxed_positive	0.21276	8.21057
811	0.174,08.28,1.94,32.0,0.081,38,boxed_positive	0.23167	8.21057
106	0.189,19.14,1.92,30.8,0.038,31,boxed_positive	0.20376	8.21057
846	1.167,19.17,1.94,25.0,0.402,33,boxed_positive	0.24370	8.18176
138	0.158,19.23,1.9,38.0,0.031,51,boxed_positive	0.21401	8.21057
301	0.144,08.03,1.90,31.0,0.425,35,boxed_positive	0.21278	8.21076
398	0.161,19.17,0.34,9.0,0.121,25,boxed_positive	0.21186	8.21076
414	0.158,08.38,1.87,34.0,0.034,21,boxed_positive	0.20087	8.21076
271	0.148,08.38,1.90,42.0,0.431,34,boxed_positive	0.24369	8.20037
283	1.128,08.48,1.94,40.0,0.013,24,boxed_positive	0.24642	8.20037
426	0.178,08.42,2.90,42.0,0.399,24,boxed_positive	0.2874	8.20037
328	1.322,04.32,1.90,35.0,0.092,30,boxed_positive	0.20160	8.20037
19	1.114,19.38,0.34,0.0,0.029,52,boxed_positive	0.2094	8.18188
291	0.107,02.38,14.36,0.0,757.23,boxed_positive	0.18447	8.18188
226	1.113,04.38,0.33,0.0,0.042,21,boxed_positive	0.20420	8.18447
448	0.140,04.37,04.38,0.0,0.122,boxed_positive	0.20025	8.18447
708	2.81,04.32,0.98,38.0,0.074,23,boxed_positive	0.23091	8.18447
70	2.108,08.28,0.38,0.0,0.007,20,boxed_positive	0.21537	8.18084
338	2.124,08.28,2.92,32.0,0.075,30,boxed_positive	0.21532	8.18796
641	0.158,19.28,1.80,30.0,0.048,37,boxed_positive	0.20278	8.18796
108	1.125,19.48,1.87,35.0,0.062,26,boxed_positive	0.24713	8.18188
418	0.128,04.28,1.95,30.0,0.219,38,boxed_positive	0.22481	8.2048
387	0.171,08.48,0.34,1.0,1.98,22,boxed_positive	0.22749	8.20023
38	0.801,04.42,0.38,0.0,0.003,37,boxed_positive	0.0436	8.20023
608	0.121,08.38,1.81,34.0,0.003,33,boxed_positive	0.24403	8.2004
31	0.158,19.38,2.40,31.0,0.051,26,boxed_positive	0.24824	8.2137
478	2.105,08.48,1.81,33.7,0.111,38,boxed_positive	0.2318	8.21028
302	0.158,19.24,0.07,4.0,0.059,36,boxed_positive	0.21778	8.21057
187	0.107,02.13,08.02,0.0,0.018,23,boxed_positive	0.20389	8.22114
138	1.117,08.34,1.91,34.0,0.401,40,boxed_positive	0.24880	8.22086
198	0.148,08.17,1.91,30.0,0.11,29,boxed_positive	0.21798	8.22167
327	0.112,19.18,0.01,0.0,0.187,36,boxed_positive	0.24198	8.21018
388	0.108,04.33,0.25,21.0,0.081,38,boxed_positive	0.20660	8.22048

- Classes to clusters evaluation

=== Run information ===

Scheme: weka.clusterers.OPTICS -E 0.9 -M 6 -A "weka.core.EuclideanDistance -R first-last" -db-output .

Relation: pima_diabetes

Instances: 768

Attributes: 9

preg

plas

pres

skin

insu

mass

pedi

age

Ignored:

class

Test mode: Classes to clusters evaluation on training data

=== Clustering model (full training set) ===

OPTICS clustering results

=====

Clustered DataObjects: 768

Number of attributes: 8

Epsilon: 0.9; minPoints: 6

Write results to file: no

Distance-type:

Number of generated clusters: 0

Elapsed time: .14

(0.) 6,148,72,35,0,33.6,0.627,50	--> c_dist: 0.237	r_dist: UNDEFINED
(285.) 7,136,74,26,135,26,0.647,51	--> c_dist: 0.212	r_dist: 0.237
(14.) 5,166,72,19,175,25.8,0.587,51	--> c_dist: 0.258	r_dist: 0.212
(603.) 7,150,78,29,126,35.2,0.692,54	--> c_dist: 0.21	r_dist: 0.212
(236.) 7,181,84,21,192,35.9,0.586,51	--> c_dist: 0.239	r_dist: 0.21
(670.) 6,165,68,26,168,33.6,0.631,49	--> c_dist: 0.223	r_dist: 0.21
(516.) 9,145,88,34,165,30.3,0.771,53	--> c_dist: 0.263	r_dist: 0.21
.		
.		
.		
(4.) 0,137,40,35,168,43.1,2.288,33	--> c_dist: 0.53	r_dist: 0.459
(357.) 13,129,0,30,0,39.9,0.569,44	--> c_dist: 0.552	r_dist: 0.479
(13.) 1,189,60,23,846,30.1,0.398,59	--> c_dist: 0.703	r_dist: 0.481
(453.) 2,119,0,0,0,19.6,0.832,72	--> c_dist: 0.632	r_dist: 0.499
(228.) 4,197,70,39,744,36.7,2.329,31	--> c_dist: 0.776	r_dist: 0.588
(579.) 2,197,70,99,0,34.7,0.575,62	--> c_dist: 0.771	r_dist: 0.646

Time taken to build model (full training data) : 0.25 seconds

Key	DataSheet	Card Distance	Transf.ability Distance
1	0.148 72 35.3 30.0 0.0 0.0 0.0 0.0	0.23738	0.23738
208	7.138 74.38 135.36 0.0 0.0 0.0 0.0	0.23451	0.23738
14	0.108 72 18.175 35.0 0.0 0.0 0.0 0.0	0.25103	0.25103
303	7.138 74.38 135.36 0.0 0.0 0.0 0.0	0.26068	0.27151
136	7.138 74.38 135.36 0.0 0.0 0.0 0.0	0.25091	0.25091
478	0.105 98.28 168.35 0.0 0.0 0.0 0.0	0.22574	0.20866
144	0.148 72 35.3 30.0 0.0 0.0 0.0 0.0	0.26238	0.25988
391	0.105 98.28 168.35 0.0 0.0 0.0 0.0	0.25968	0.25103
408	8.708 72 35.3 30.0 0.0 0.0 0.0 0.0	0.24818	0.27151
111	0.108 72 35.3 30.0 0.0 0.0 0.0 0.0	0.18818	0.18818
131	0.108 72 35.3 30.0 0.0 0.0 0.0 0.0	0.25018	0.19818
388	0.118 74.38 135.36 0.0 0.0 0.0 0.0	0.27418	0.18818
317	0.148 72 35.3 30.0 0.0 0.0 0.0 0.0	0.22981	0.18818
378	0.108 72 35.3 30.0 0.0 0.0 0.0 0.0	0.18003	0.18818
418	1.128 64.28 115.38 0.0 0.0 0.0 0.0	0.17254	0.18003
331	2.148 64.28 115.38 0.0 0.0 0.0 0.0	0.18808	0.17254
384	1.128 64.28 115.38 0.0 0.0 0.0 0.0	0.18713	0.17254
307	0.107 68.14 148.24 0.0 0.0 0.0 0.0	0.18131	0.18713
328	1.127 72.21 168.25 0.0 0.0 0.0 0.0	0.18152	0.19131
308	1.115 63.12 168.24 0.0 0.0 0.0 0.0	0.17735	0.18131
633	1.128 64.28 115.38 0.0 0.0 0.0 0.0	0.18328	0.18131
341	0.108 62.17 202.20 1.0 0.0 0.0 0.0	0.18	0.18131
448	0.128 74.18 63.30 0.0 0.0 0.0 0.0	0.18513	0.18713
143	1.108 64.28 115.38 0.0 0.0 0.0 0.0	0.18517	0.18152
3	1.89 62.33 04.28 1.0 1.0 0.0 0.0	0.18648	0.18717
158	2.88 74.18 63.30 0.0 0.0 0.0 0.0	0.1908	0.18648
348	1.14 64.28 115.38 0.0 0.0 0.0 0.0	0.11548	0.18648
302	2.108 64.28 115.38 0.0 0.0 0.0 0.0	0.12481	0.17488
367	1.81 70.31 0.30 0.0 0.0 0.0 0.0	0.12117	0.17488
383	0.101 68.38 202.20 0.0 0.0 0.0 0.0	0.15584	0.17488
397	2.02 62.20 71.0 0.0 0.0 0.0 0.0	0.1544	0.17488
668	1.81 64.28 115.38 0.0 0.0 0.0 0.0	0.14451	0.17488
208	1.80 64.27 37.20 0.0 0.0 0.0 0.0	0.13188	0.17488
334	1.80 64.28 115.38 0.0 0.0 0.0 0.0	0.13003	0.17488
387	0.101 64.17 212.21 0.0 0.0 0.0 0.0	0.12484	0.17003
380	1.80 64.28 115.38 0.0 0.0 0.0 0.0	0.11775	0.17003
128	1.87 64.19 63.30 0.0 0.0 0.0 0.0	0.13388	0.17003
458	1.82 64.17 202.20 0.0 0.0 0.0 0.0	0.13188	0.17175
138	1.108 74.18 63.30 0.0 0.0 0.0 0.0	0.14538	0.17175
303	2.89 70.35 44.28 0.0 0.0 0.0 0.0	0.12943	0.17175
103	1.80 62.24 44.28 0.0 0.0 0.0 0.0	0.15384	0.17003
137	0.01 63.63 202.20 0.0 0.0 0.0 0.0	0.14438	0.17004
441	3.81 66.33 202.20 0.0 0.0 0.0 0.0	0.15514	0.17004
218	2.87 63.20 127.70 0.0 0.0 0.0 0.0	0.13003	0.17004
198	2.80 70.31 0.37 0.0 0.0 0.0 0.0	0.14003	0.17003

Post-lab Questions:

1. What is an outlier?

Ans:

- a. In statistics, an outlier is a data point that differs significantly from other observations.
- b. An outlier may be due to variability in the measurement or it may indicate experimental error; the latter are sometimes excluded from the data set. An outlier can cause serious problems in statistical analyses.
- c. Sometimes outliers might be errors that we want to exclude or an anomaly that we don't want to include in our analysis.
- d. But at other times it can reveal insights into special cases in our data that we may not otherwise notice.
- e. Outliers can be of two types: Univariate and Multivariate.
- f. Univariate outliers are outliers in a 1 dimensional space.
- g. Multivariate outliers are outliers in an n-dimensional space.
- h. Most common causes of outliers on a data set:
 - Data entry errors (human errors)
 - Measurement errors (instrument errors)
 - Experimental errors (data extraction or experiment planning/executing errors)
 - Intentional (dummy outliers made to test detection methods)
 - Data processing errors (data manipulation or data set unintended mutations)
 - Sampling errors (extracting or mixing data from wrong or various sources)
 - Natural (not an error, novelties in data)

2. Give any method to detect an outlier.

Ans:

- a. Most commonly used method to detect outliers is visualization.
- b. We use various visualization methods, like Box-plot, Histogram, Scatter Plot
- c. Use capping methods. Any value which out of range of 5th and 95th percentile can be considered as outlier 3.Data points, three or more standard deviation away from mean are considered outlier
- d. d.Outlier detection is merely a special case of the examination of data for influential datapoint sanditals depends on the business understanding
- e. Bivariate and multivariate outliers are typically measured using either an index of influence or leverage, or distance. Some of the most popular methods for outlier detection are:

Z-Score or Extreme Value Analysis (parametric)

Probabilistic and Statistical Modeling (parametric)

Linear Regression Models (PCA, LMS)

Proximity Based Models(non-parametric)

Information Theory Models

High Dimensional Outlier Detection Methods (high dimensional sparse data)

Z-Score

- a. The z-score or standard score of an observation is a metric that indicates how many standard deviations a data point is from the sample's mean, assuming a gaussian distribution. This makes z-score a parametric method. Very frequently data points are not to described by a gaussian distribution, this problem can be solved by applying transformations to data i.e.: scaling it.
- b. SomePython libraries likeSciPy and Sci-kit Learn have easy to use functions and classes for easy implementation along with Pandas andNumPy.

- c. After making the appropriate transformations to the selected feature space of the dataset, the z-score of any data point can be calculated with the following expression: $Z = \frac{x - \mu}{\sigma}$
- d. When computing the z-score for each sample on the dataset a threshold must be specified. Some 'Good Thumbrule' thresholds can be: 2.5, 3, 3.5 or more standard deviations.
- e. By 'tagging' or removing the data points that lay beyond a given threshold We are classifying data into outliers and not outliers
- f. Z-score is a simple, yet powerful method to get rid of outliers in data if are dealing with parametric distributions in a low dimensional feature space