University of Colombo School of Computing

Bachelor of Information Systems

Data Analytics

IS 4003

Assignment – Data Mining

E.H.Grero
15020258
2015/IS/025

Objectives

The main objective of this dataset is to measure geometrical properties of kernels belonging to three different varieties of wheat namely Kama, Rosa and Canadian. A soft X-ray technique and GRAINS package is been used to construct all seven, real-valued attributes.

Benefit of using association rule mining would be to identify the inter relationships of the geometrical properties and identify separately the properties that will be able to identify the three different varieties of wheat Kama, Rosa and Canadian separately and clearly.

Dataset description

The dataset was taken from the UCI machine learning repository.

URL of the dataset :- https://archive.ics.uci.edu/ml/datasets/seeds

The source of the dataset is taken from MaÅ, gorzata Charytanowicz, Jerzy Niewczas from Institute of Mathematics and Computer Science and Piotr Kulczycki, Piotr A. Kowalski, Szymon Lukasik, Slawomir Zak from Department of Automatic Control and Information Technology

To construct the data, seven geometric parameters of wheat kernels have been measured:

1. area A,
2. perimeter P,
3. compactness C = 4*pi*A/P^2,
4. length of kernel,
5. width of kernel,
6. asymmetry coefficient
7. length of kernel groove.

All these parameters are real-valued continuous.

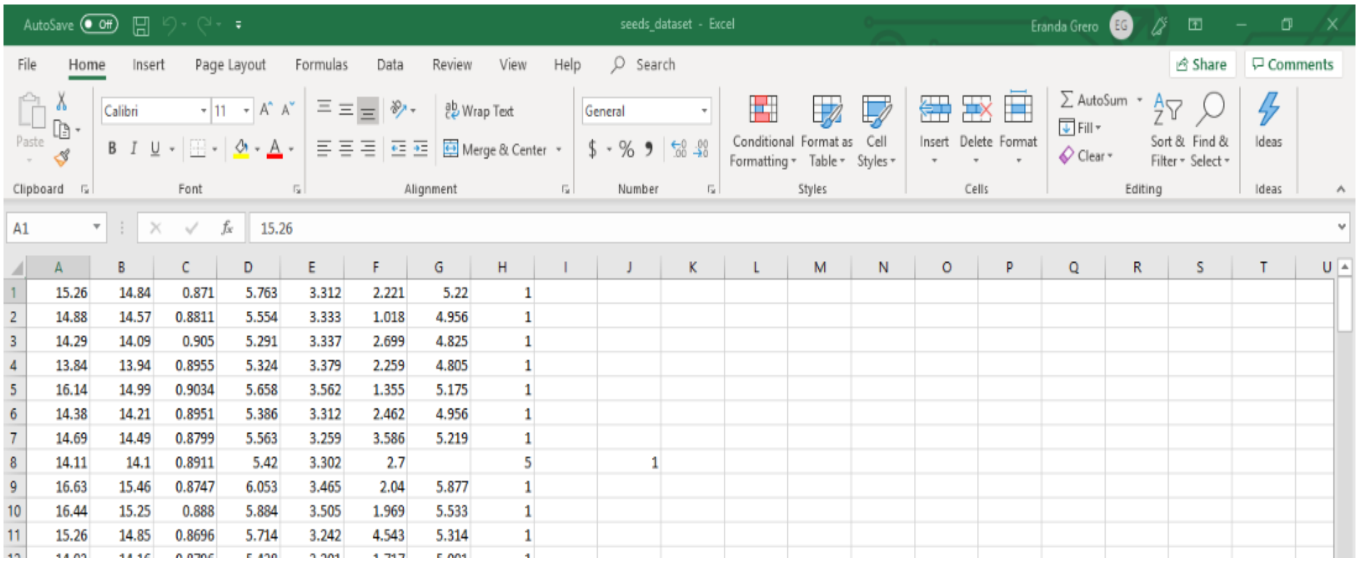
The examined group or the data set comprised kernels belonging to three different varieties of wheat: Kama, Rosa and Canadian, 70 elements each, randomly selected for the experiment. High quality visualization of the internal kernel structure has been detected using a soft X-ray technique. It is non-destructive and considerably cheaper than other more sophisticated imaging techniques like scanning microscopy or laser technology. The images were recorded on 13x18 cm X-ray KODAK plates. Studies were conducted using combine harvested wheat grain originating from experimental fields, explored at the Institute of Agrophysics of the Polish Academy of Sciences in Lublin.

Reason for selecting the dataset

The dataset contains 210 instances. 70 each from the three varieties of wheat namely Kama, Rosa and Canadian. Therefore, the dataset contains enough instances to identify association rules and patterns. The data set can be used for the tasks of classification and cluster analysis also made me choose this dataset.

Preprocessing

Dataset when downloaded from the UCI machine repository was a text file. First the text file was open from Microsoft excel and it is shown as below.

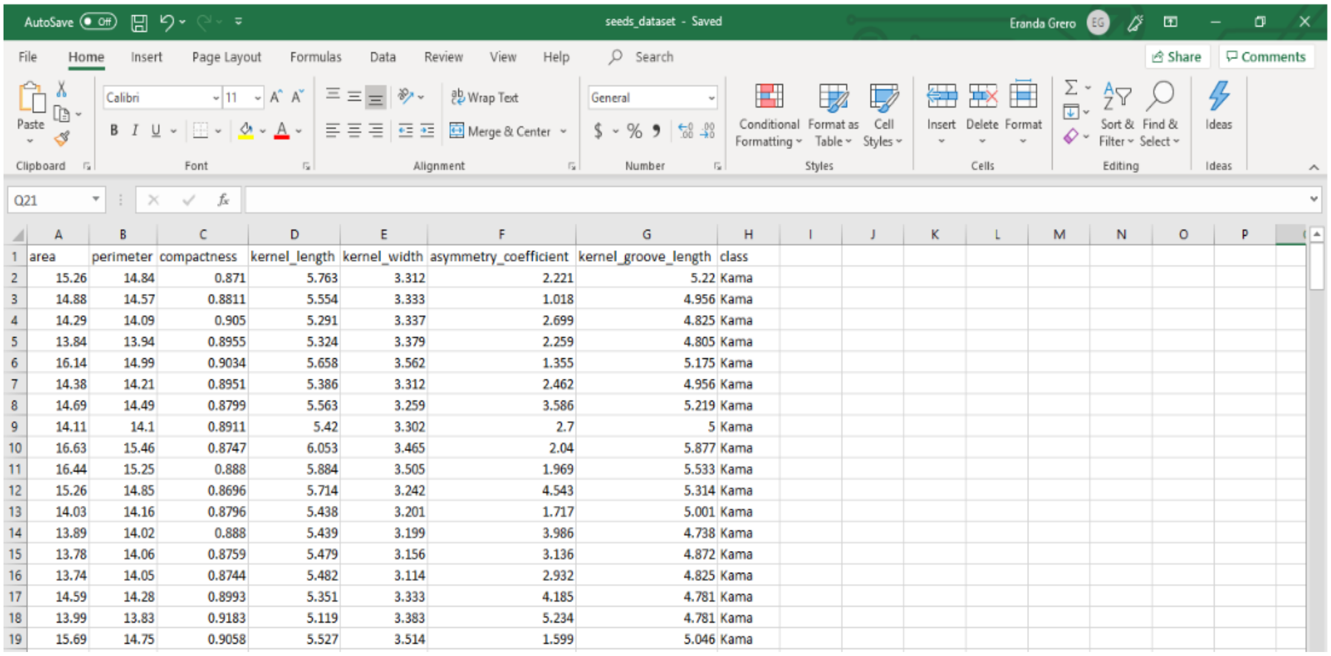


A1 15.26

	A	B	C	D	E	F	G	H	I	J
1	15.26	14.84	0.871	5.763	3.312	2.221	5.22	1		
2	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1		
3	14.29	14.09	0.905	5.291	3.337	2.699	4.825	1		
4	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1		
5	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1		
6	14.38	14.21	0.8951	5.386	3.312	2.462	4.956	1		
7	14.69	14.49	0.8799	5.563	3.259	3.586	5.219	1		
8	14.11	14.1	0.8911	5.42	3.302	2.7		5		1
9	16.63	15.46	0.8747	6.053	3.465	2.04	5.877	1		
10	16.44	15.25	0.888	5.884	3.505	1.969	5.533	1		
11	15.26	14.85	0.8696	5.714	3.242	4.543	5.314	1		

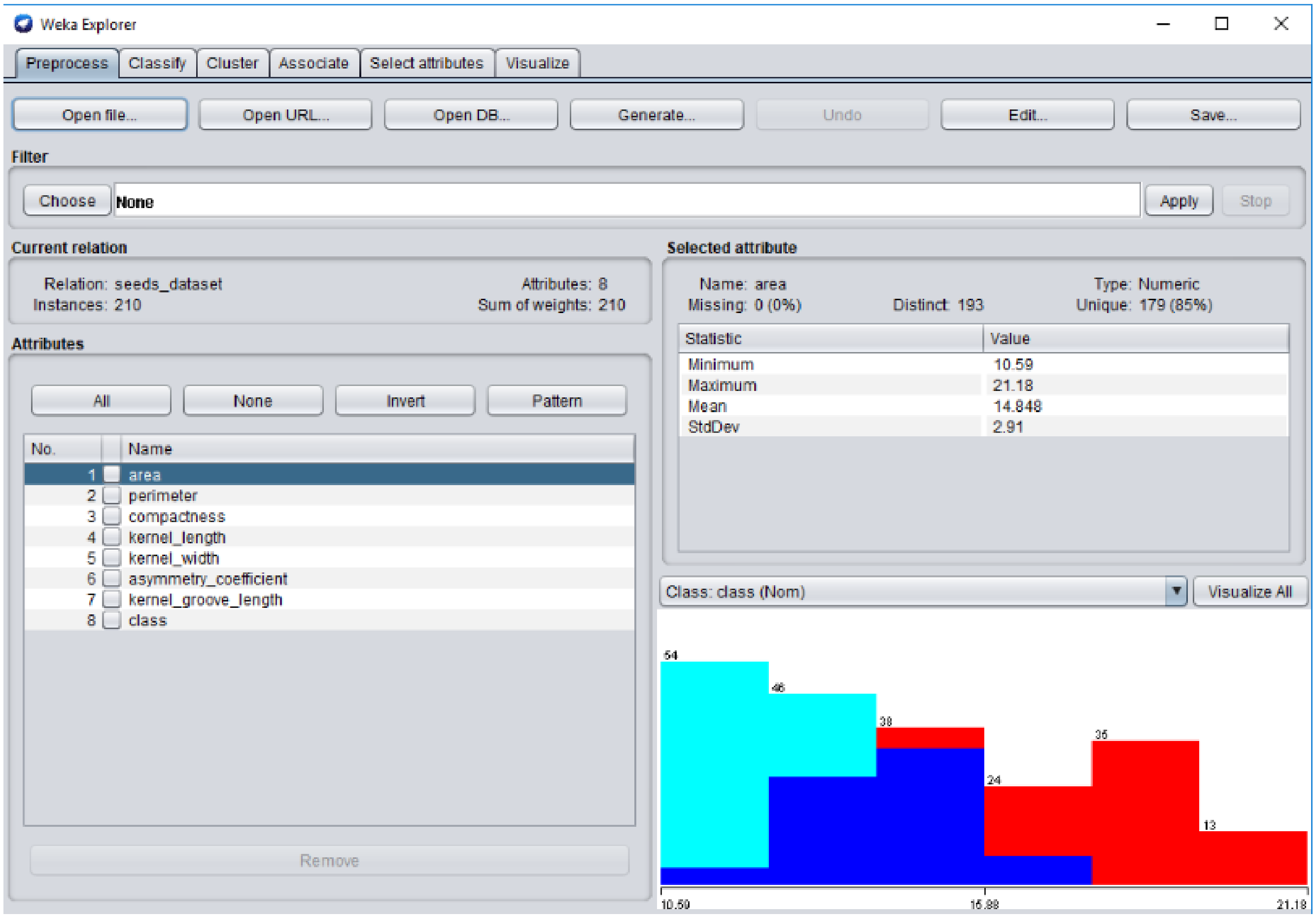
The data set did not contain attribute names and it had no missing values, but some data cells have shifted to the right as shown in the figure. All the values in the dataset were numeric.

After arranging the data set by making the data rows which have shifted to the right and putting the attribute names the dataset looks like below. In the class field, the values 1,2,3 were changed to Kama, Rosa and Canadian for proper visibility of association rules and clustering.



	A	B	C	D	E	F	G	H
1	area	perimeter	compactness	kernel_length	kernel_width	asymmetry_coefficient	kernel_groove_length	class
2	15.26	14.84	0.871	5.763	3.312	2.221	5.22	Kama
3	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	Kama
4	14.29	14.09	0.905	5.291	3.337	2.699	4.825	Kama
5	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	Kama
6	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	Kama
7	14.38	14.21	0.8951	5.386	3.312	2.462	4.956	Kama
8	14.69	14.49	0.8799	5.563	3.259	3.586	5.219	Kama
9	14.11	14.1	0.8911	5.42	3.302	2.7	5	Kama
10	16.63	15.46	0.8747	6.053	3.465	2.04	5.877	Kama
11	16.44	15.25	0.888	5.884	3.505	1.969	5.533	Kama
12	15.26	14.85	0.8696	5.714	3.242	4.543	5.314	Kama
13	14.03	14.16	0.8796	5.438	3.201	1.717	5.001	Kama
14	13.89	14.02	0.888	5.439	3.199	3.986	4.738	Kama
15	13.78	14.06	0.8759	5.479	3.156	3.136	4.872	Kama
16	13.74	14.05	0.8744	5.482	3.114	2.932	4.825	Kama
17	14.59	14.28	0.8993	5.351	3.333	4.185	4.781	Kama
18	13.99	13.83	0.9183	5.119	3.383	5.234	4.781	Kama
19	15.69	14.75	0.9058	5.527	3.514	1.599	5.046	Kama

The dataset was then saved in .csv format and opened using Weka as shown below.


Weka Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose
Apply Stop
All None Invert Pattern
Remove
Visualize All
54
46
38
24
36
13
10.59
15.88
21.18

Filter: None

Current relation

Relation: seeds_dataset
Instances: 210
Attributes: 8
Sum of weights: 210

Attributes

No.	Name
1	area
2	perimeter
3	compactness
4	kernel_length
5	kernel_width
6	asymmetry_coefficient
7	kernel_groove_length
8	class

Selected attribute

Name: area
Missing: 0 (0%)
Distinct: 193
Type: Numeric
Unique: 179 (85%)

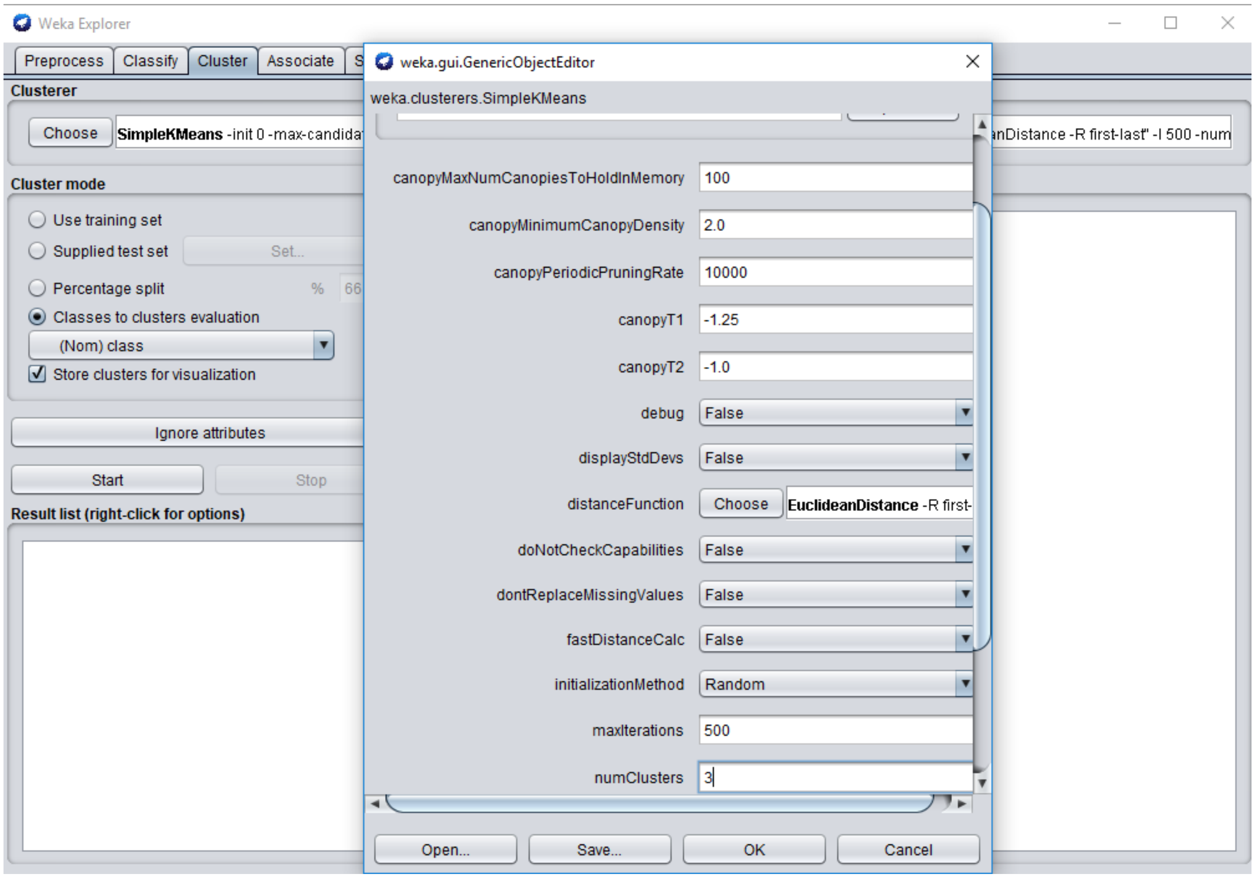
Statistic	Value
Minimum	10.59
Maximum	21.18
Mean	14.848
StdDev	2.91

Class: class (Nom)

The csv file was then saved in .arff format.

The seeds_data.arff file was then loaded into weka and before doing the associate rule mining I did the clustering using k-means (SimpleKMeans), as after discretization clustering cannot be done.

As shown below the number of clusters were given as 3 and the Cluster mode was selected as Classes to clusters evaluation.



weka.gui.GenericObjectEditor

Clusterer

Choose SimpleKMeans -init 0 -max-candida ... anDistance -R first-last" -I 500 -num

Cluster mode

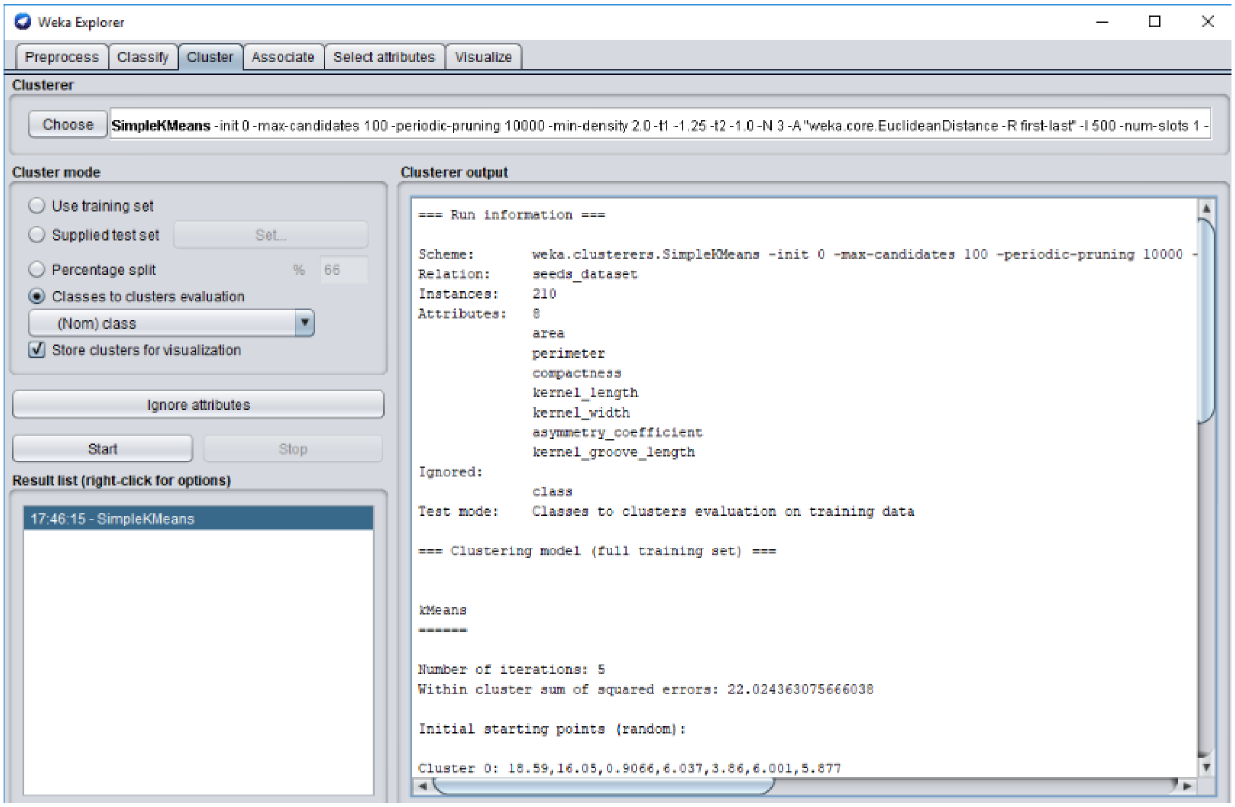
○ Use training set
○ Supplied test set Set...
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

weka.clusterers.SimpleKMeans

canopyMaxNumCanopiesToHoldInMemory	100
canopyMinimumCanopyDensity	2.0
canopyPeriodicPruningRate	10000
canopyT1	-1.25
canopyT2	-1.0
debug	False
displayStdDevs	False
distanceFunction	Choose EuclideanDistance -R first-
doNotCheckCapabilities	False
dontReplaceMissingValues	False
fastDistanceCalc	False
initializationMethod	Random
maxIterations	500
numClusters	3

Then clicked OK and then clicked Start. Output of the k-means clustering can be seen below.

(Refer ‘Kmeans-original.txt’ for more details)

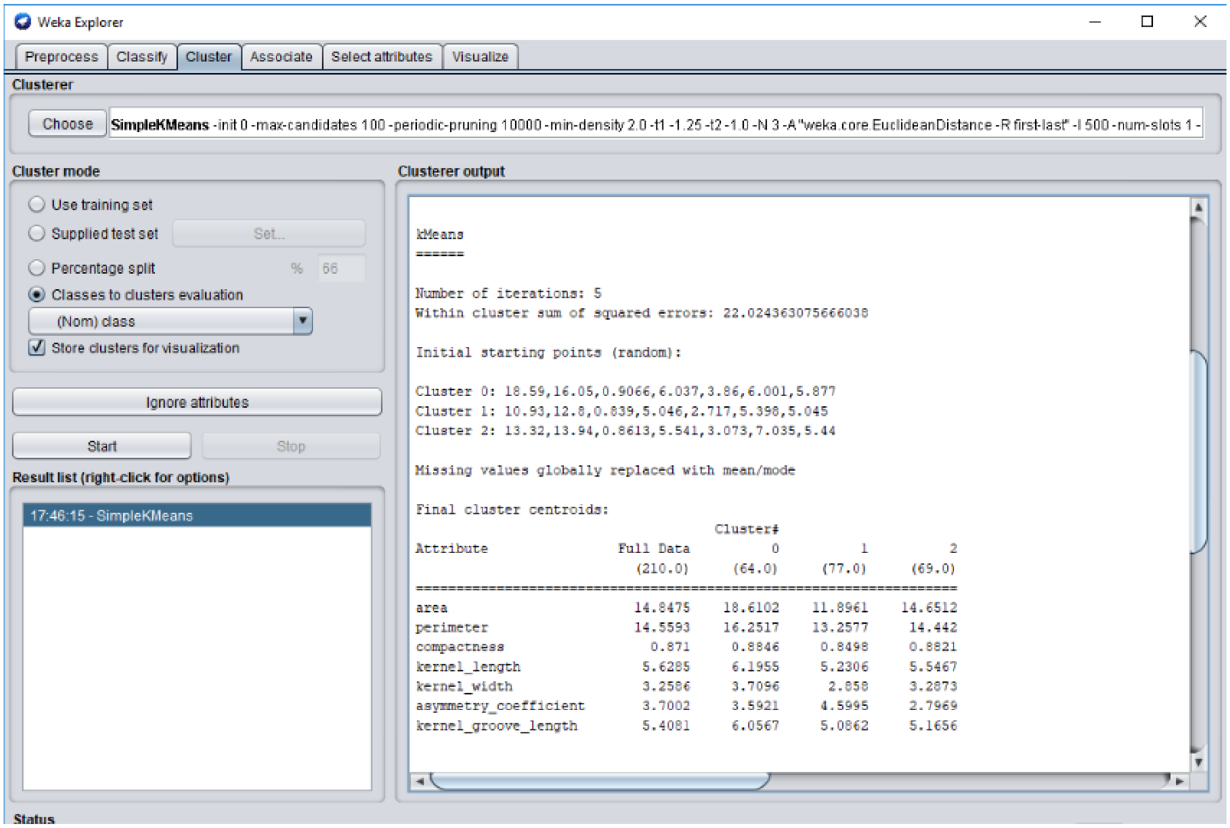


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -

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

17:46:15 - SimpleKMeans

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -
Relation:     seeds_dataset
Instances:    210
Attributes:   8
              area
              perimeter
              compactness
              kernel_length
              kernel_width
              asymmetry_coefficient
              kernel_groove_length
Ignored:
              class
Test mode:    Classes to clusters evaluation on training data

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


kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 22.024363075666038

Initial starting points (random):

Cluster 0: 18.59,16.05,0.9066,6.037,3.86,6.001,5.877
```

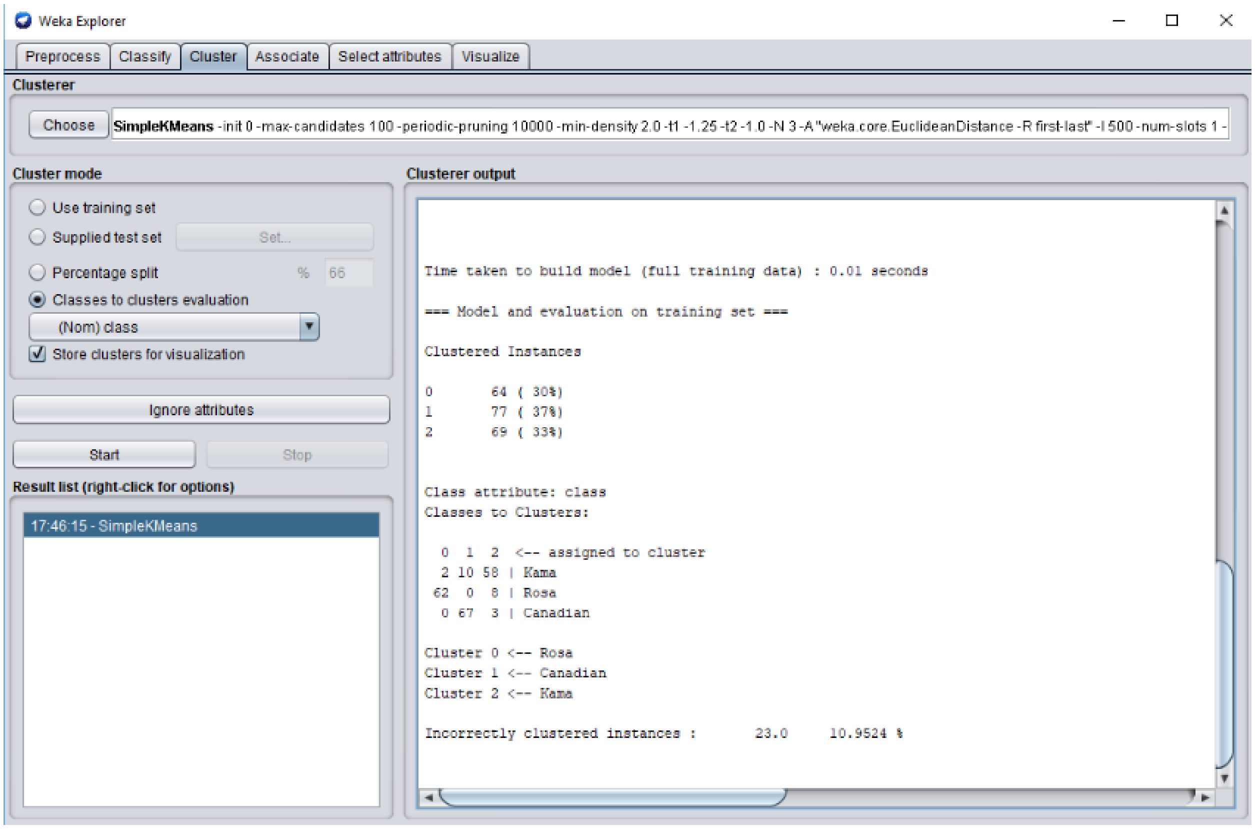


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -

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

17:46:15 - SimpleKMeans

```

kMeans
======

Number of iterations: 5
Within cluster sum of squared errors: 22.024363075666038

Initial starting points (random):

Cluster 0: 18.59,16.05,0.9066,6.037,3.86,6.001,5.877
Cluster 1: 10.93,12.8,0.839,5.046,2.717,5.398,5.045
Cluster 2: 13.32,13.94,0.8613,5.541,3.073,7.035,5.44

Missing values globally replaced with mean/mode

Final cluster centroids:
                                     Cluster#
Attribute                Full Data          0          1          2
                           (210.0)     (64.0)     (77.0)     (69.0)
===================================================================
area                       14.8475    18.6102    11.8961    14.6512
perimeter                  14.5593    16.2517    13.2577     14.442
compactness                  0.871     0.8846     0.8498     0.8821
kernel_length               5.6285     6.1955     5.2306     5.5467
kernel_width                3.2586     3.7096      2.858     3.2873
asymmetry_coefficient       3.7002     3.5921     4.5995     2.7969
kernel_groove_length        5.4081     6.0567     5.0862     5.1656
```



Weka Explorer

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 -`

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)

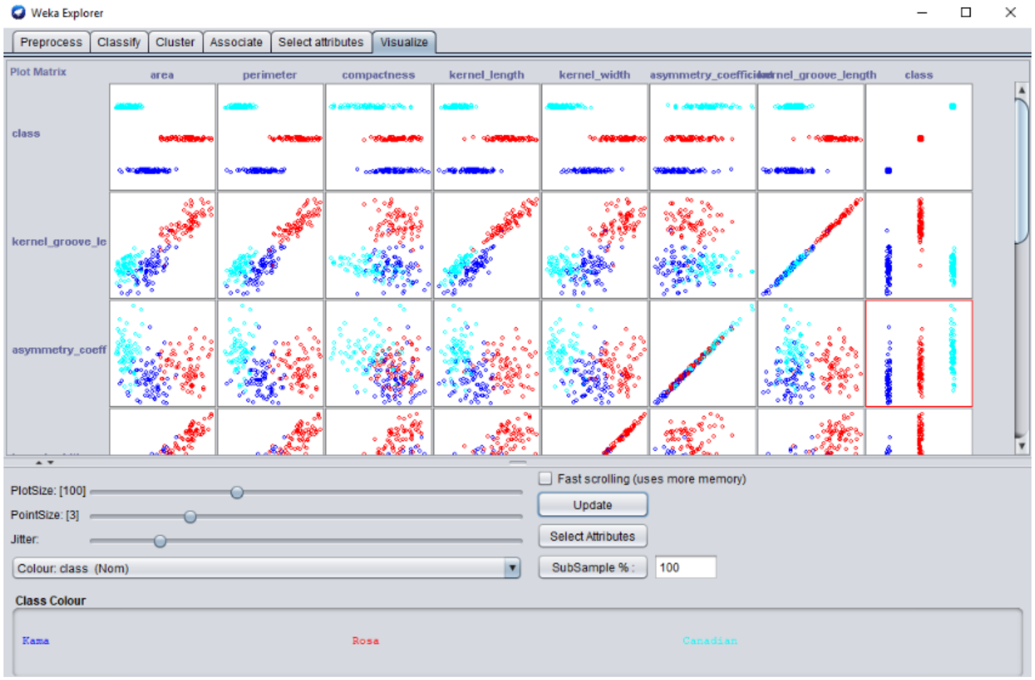
17:46:15 - SimpleKMeans

Clusterer output

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

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

Clustered Instances

0       64 ( 30%)
1       77 ( 37%)
2       69 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  2 10 58 | Kama
 62  0  8 | Rosa
  0 67  3 | Canadian

Cluster 0 <-- Rosa
Cluster 1 <-- Canadian
Cluster 2 <-- Kama

Incorrectly clustered instances :       23.0      10.9524 %
```

As seen from the figures the Incorrectly clustered instances are 23 and the percentage is 10.9524%.

Clustering can be visualized as shown below.



Weka Explorer

☐ Fast scrolling (uses more memory)

PlotSize: [100]

PointSize: [3]

Jitter:

Colour: class (Nom)

SubSample % : 100

Class Colour

Kama Rosa Canadian

Then by ignoring attributes I tried to take the most accurate clustering with least incorrect instances.

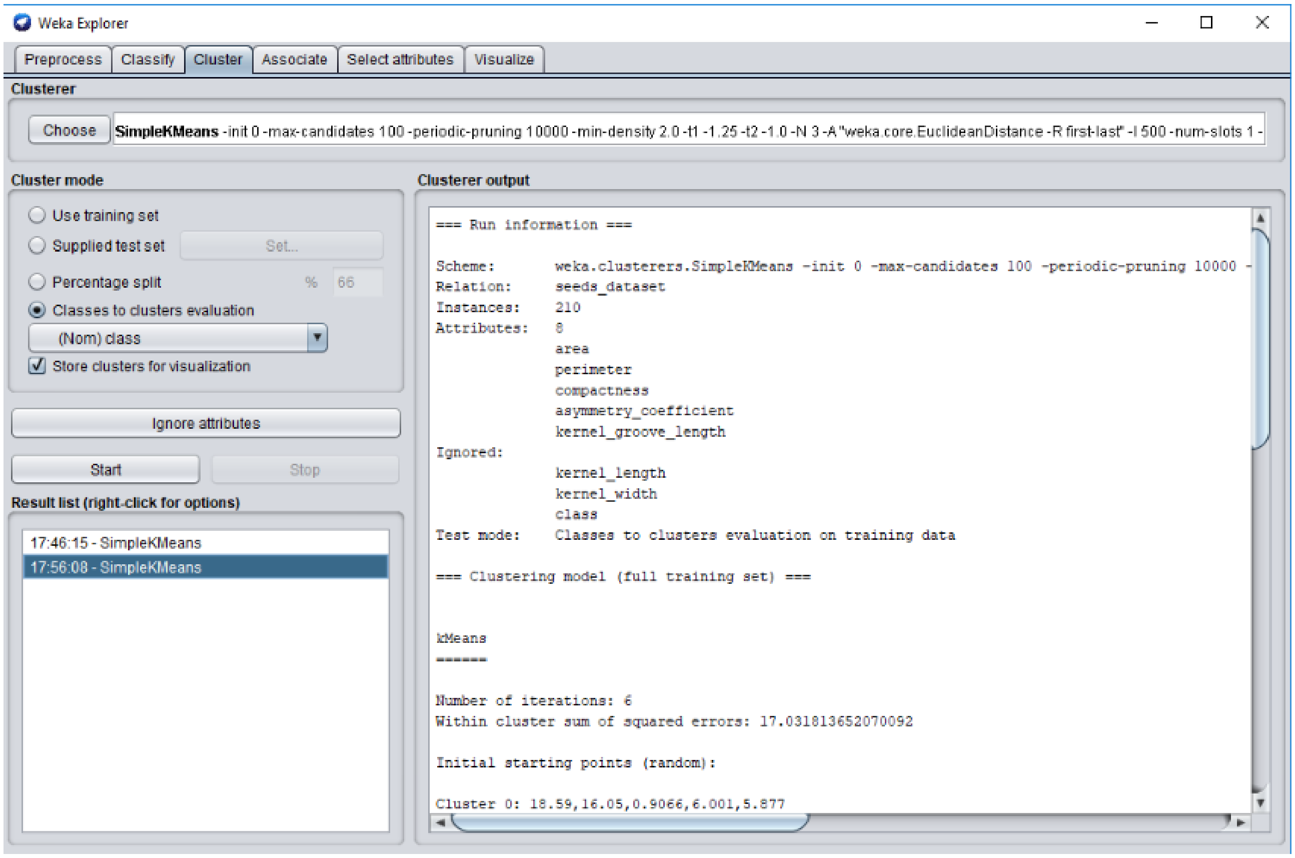
Two results showed the minimum incorrect clustered instances as only 14 and the percentage as 6.6667%. One result was by ignoring the attributes kernel length and kernel width and the other result I got from ignoring area and kernel width as shown below.

Reasons for ignoring kernel width and kernel length in one result and then area and kernel width in another result was by looking at the point spread by visualize I identified according to their spread of points they should be ignored.

I tried for all other combinations as well to verify the minimum incorrectly clustered instances.

Result 1 (Ignoring kernel_length and kernel_width)

(Refer 'Kmeans-result1.txt' for more details)

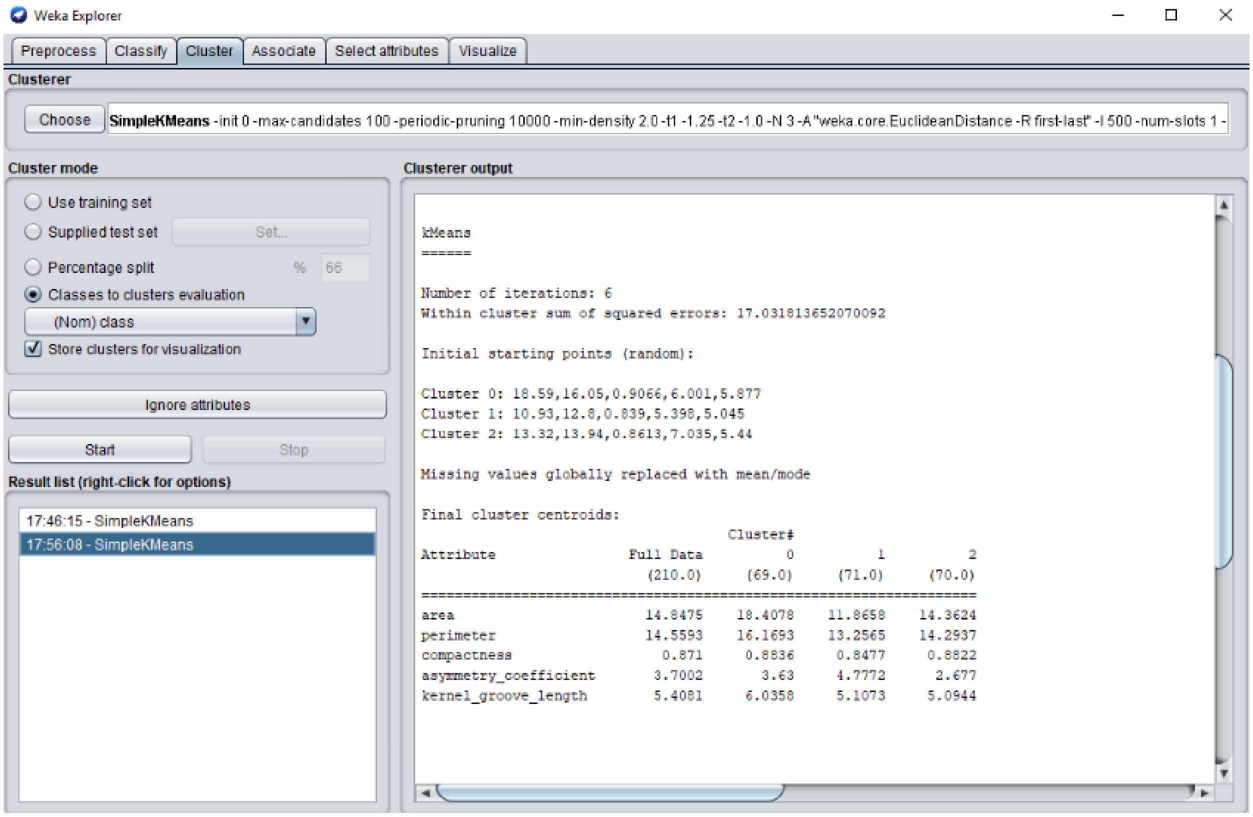


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -

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

17:46:15 - SimpleKMeans
17:56:08 - SimpleKMeans

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -
Relation:     seeds_dataset
Instances:    210
Attributes:   8
              area
              perimeter
              compactness
              asymmetry_coefficient
              kernel_groove_length
Ignored:
              kernel_length
              kernel_width
              class
Test mode:    Classes to clusters evaluation on training data

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


kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 17.031813652070092

Initial starting points (random):

Cluster 0: 18.59,16.05,0.9066,6.001,5.877
```



Clusterer

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 -

Cluster mode

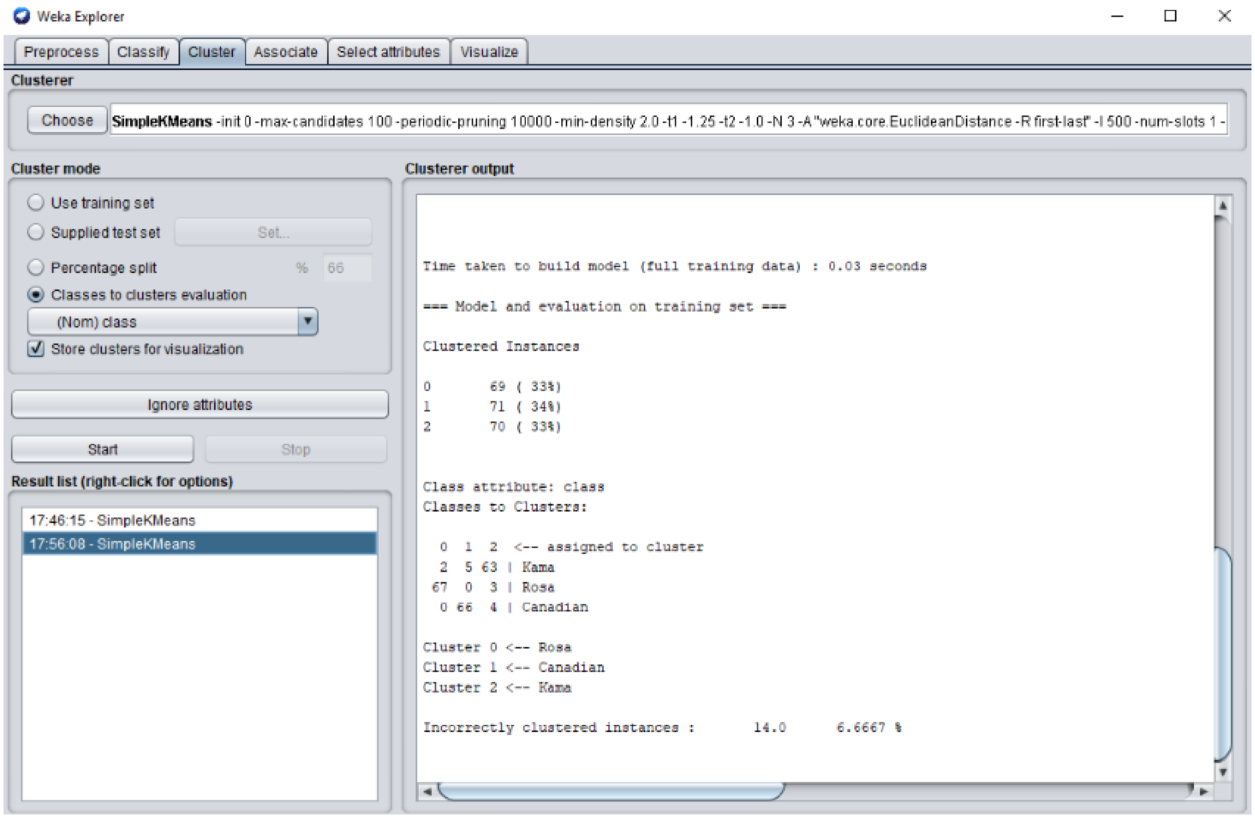
○ Use training set
○ Supplied test set
○ Percentage split % 66
● Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

Result list (right-click for options)

17:46:15 - SimpleKMeans
17:56:08 - SimpleKMeans

Clusterer output

```
kMeans
======

Number of iterations: 6
Within cluster sum of squared errors: 17.031813652070092

Initial starting points (random):

Cluster 0: 18.59,16.05,0.9066,6.001,5.877
Cluster 1: 10.93,12.8,0.839,5.398,5.045
Cluster 2: 13.32,13.94,0.8613,7.035,5.44

Missing values globally replaced with mean/mode

Final cluster centroids:
                                    Cluster#
Attribute                Full Data         0          1          2
                           (210.0)    (69.0)     (71.0)     (70.0)
==================================================================
area                       14.8475   18.4078    11.8658    14.3624
perimeter                  14.5593   16.1693    13.2565    14.2937
compactness                  0.871    0.8836     0.8477     0.8822
asymmetry_coefficient       3.7002      3.63     4.7772      2.677
kernel_groove_length        5.4081    6.0358     5.1073     5.0944
```



Clusterer

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 -

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)

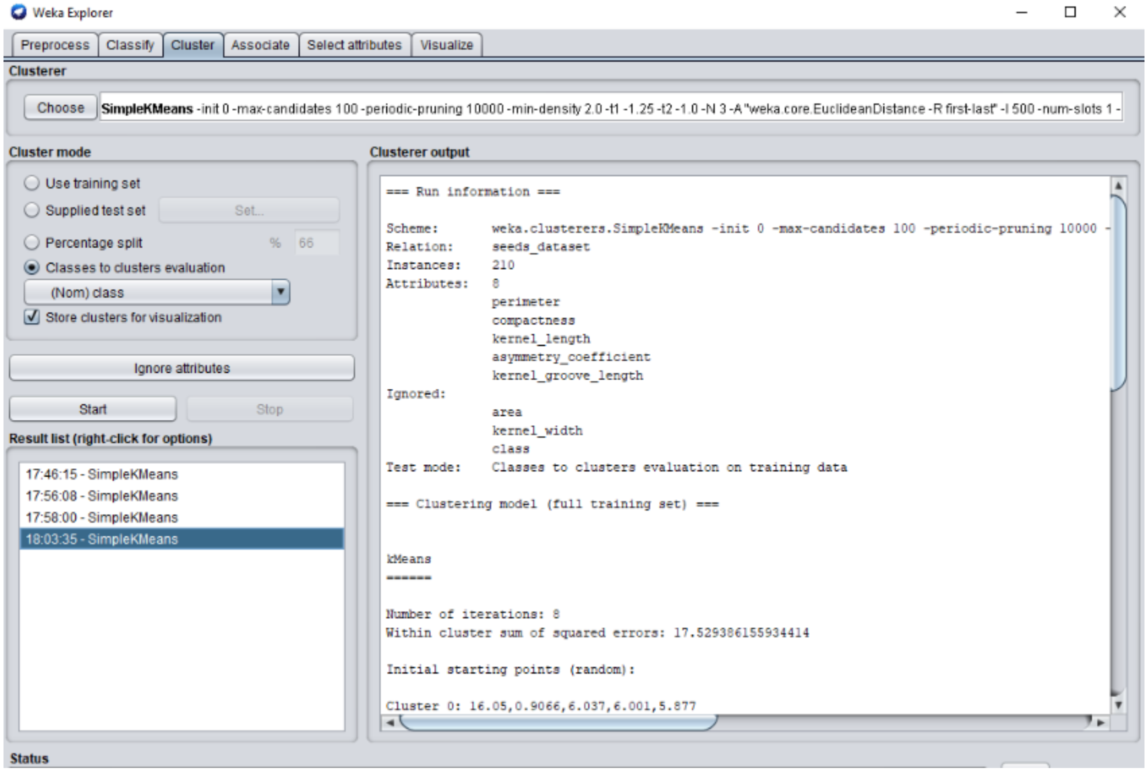
17:46:15 - SimpleKMeans
17:56:08 - SimpleKMeans

Clusterer output

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

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

Clustered Instances

0       69 ( 33%)
1       71 ( 34%)
2       70 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  2  5 63 | Kama
 67  0  3 | Rosa
  0 66  4 | Canadian

Cluster 0 <-- Rosa
Cluster 1 <-- Canadian
Cluster 2 <-- Kama

Incorrectly clustered instances :       14.0      6.6667 %
```

<u>Result 2 </u>(Ignoring area and kernel_width)

(Refer ‘Kmeans-result2.txt’ for more details)

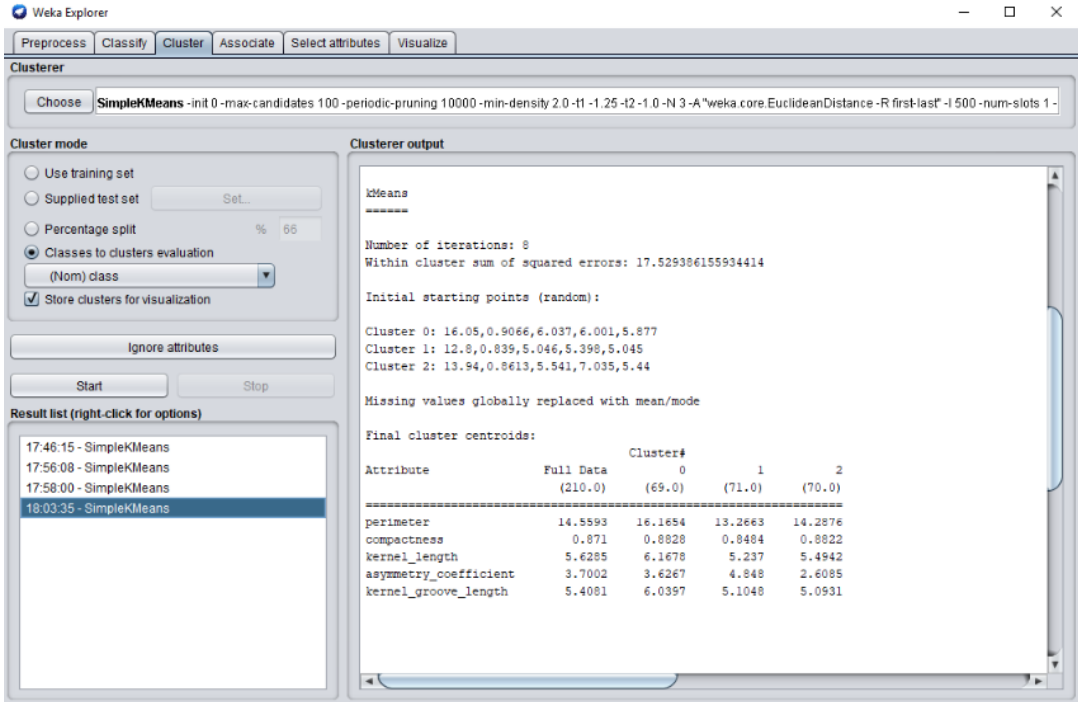


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -

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

17:46:15 - SimpleKMeans
17:56:08 - SimpleKMeans
17:58:00 - SimpleKMeans
18:03:35 - SimpleKMeans

```
=== Run information ===

Scheme:       weka.clusterers.SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -
Relation:     seeds_dataset
Instances:    210
Attributes:   8
              perimeter
              compactness
              kernel_length
              asymmetry_coefficient
              kernel_groove_length
Ignored:
              area
              kernel_width
              class
Test mode:    Classes to clusters evaluation on training data

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


kMeans
======

Number of iterations: 8
Within cluster sum of squared errors: 17.529386155934414

Initial starting points (random):

Cluster 0: 16.05,0.9066,6.037,6.001,5.877
```

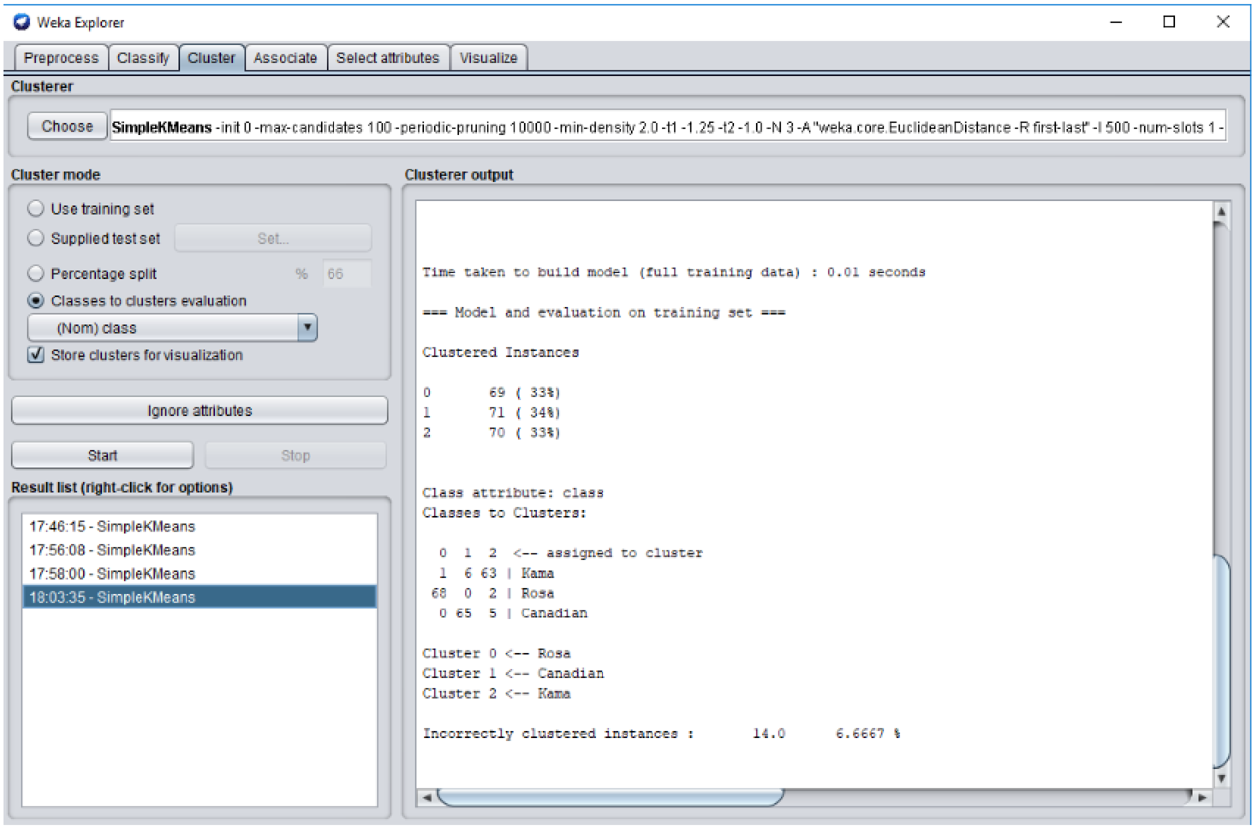


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -

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

17:46:15 - SimpleKMeans
17:56:08 - SimpleKMeans
17:58:00 - SimpleKMeans
18:03:35 - SimpleKMeans

```

kMeans
======

Number of iterations: 8
Within cluster sum of squared errors: 17.529386155934414

Initial starting points (random):

Cluster 0: 16.05,0.9066,6.037,6.001,5.877
Cluster 1: 12.8,0.839,5.046,5.398,5.045
Cluster 2: 13.94,0.8613,5.541,7.035,5.44

Missing values globally replaced with mean/mode

Final cluster centroids:
                                     Cluster#
Attribute               Full Data          0          1          2
                          (210.0)     (69.0)     (71.0)     (70.0)
==================================================================
perimeter                 14.5593    16.1654    13.2663    14.2876
compactness                 0.871     0.8828     0.8484     0.8822
kernel_length              5.6285     6.1678      5.237     5.4942
asymmetry_coefficient      3.7002     3.6267      4.848     2.6085
kernel_groove_length       5.4081     6.0397     5.1048     5.0931
```



Clusterer: 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 -

Cluster mode:

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

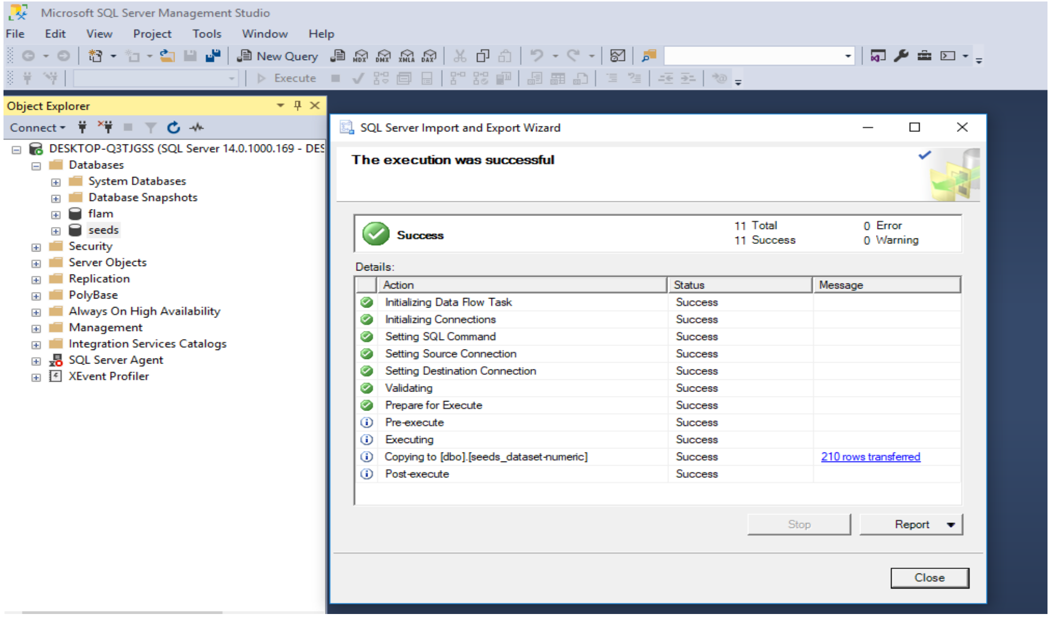
Result list:

- 17:46:15 - SimpleKMeans
- 17:56:08 - SimpleKMeans
- 17:58:00 - SimpleKMeans
- 18:03:35 - SimpleKMeans

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

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

Clustered Instances

0       69 ( 33%)
1       71 ( 34%)
2       70 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  1  6 63 | Kama
 68  0  2 | Rosa
  0 65  5 | Canadian

Cluster 0 <-- Rosa
Cluster 1 <-- Canadian
Cluster 2 <-- Kama

Incorrectly clustered instances :       14.0      6.6667 %
```

Accessing data directly from a database (JDBC)

I did also by using the JDBC connection method by creating a database in SQL server management studio and then by importing the csv file to that database as shown below.



- DESKTOP-Q3TJGSS (SQL Server 14.0.1000.169 - DES
  - Databases
    - System Databases
    - Database Snapshots
    - flam
    - seeds
  - Security
  - Server Objects
  - Replication
  - PolyBase
  - Always On High Availability
  - Management
  - Integration Services Catalogs
  - SQL Server Agent
  - XEvent Profiler

The execution was successful

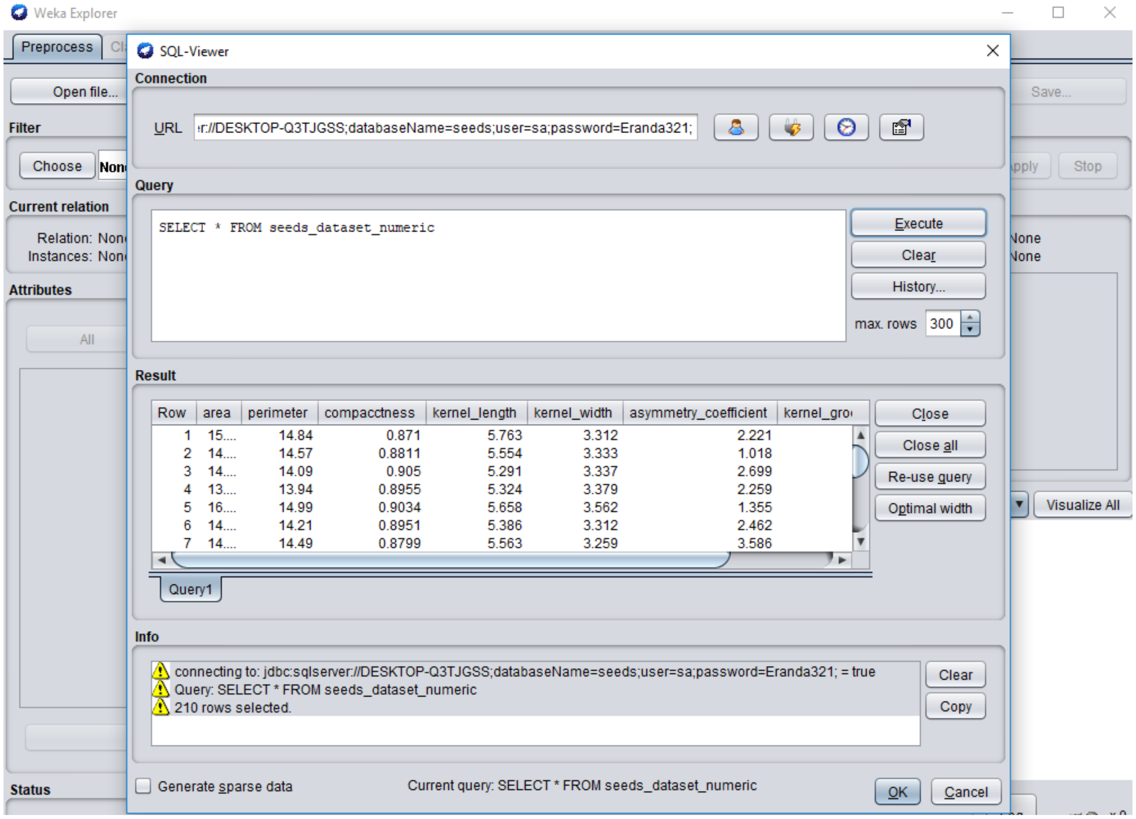
Success — 11 Total, 11 Success, 0 Error, 0 Warning

Details:

Action	Status	Message
Initializing Data Flow Task	Success	
Initializing Connections	Success	
Setting SQL Command	Success	
Setting Source Connection	Success	
Setting Destination Connection	Success	
Validating	Success	
Prepare for Execute	Success	
Pre-execute	Success	
Executing	Success	
Copying to [dbo].[seeds_dataset-numeric]	Success	210 rows transferred
Post-execute	Success	

Next JDBC driver should be downloaded and extracted. Then I edited the Database Utils.props by giving the JDBC connection database, username and password. Then enabled TCP/IP and enabled the ‘sa’ account by giving also a password and enabled SQL Authentication.

Then by using Open DB option in weka I connected to the SQL database and ran a SELECT * query to get all data from the data set.



URL r://DESKTOP-Q3TJGSS;databaseName=seeds;user=sa;password=Eranda321;

```
SELECT * FROM seeds_dataset_numeric
```

max. rows 300

Row	area	perimeter	compacctness	kernel_length	kernel_width	asymmetry_coefficient	kernel_gro
1	15....	14.84	0.871	5.763	3.312	2.221	
2	14....	14.57	0.8811	5.554	3.333	1.018	
3	14....	14.09	0.905	5.291	3.337	2.699	
4	13....	13.94	0.8955	5.324	3.379	2.259	
5	16....	14.99	0.9034	5.658	3.562	1.355	
6	14....	14.21	0.8951	5.386	3.312	2.462	
7	14....	14.49	0.8799	5.563	3.259	3.586	

```
connecting to: jdbc:sqlserver://DESKTOP-Q3TJGSS;databaseName=seeds;user=sa;password=Eranda321; = true
Query: SELECT * FROM seeds_dataset_numeric
210 rows selected.
```

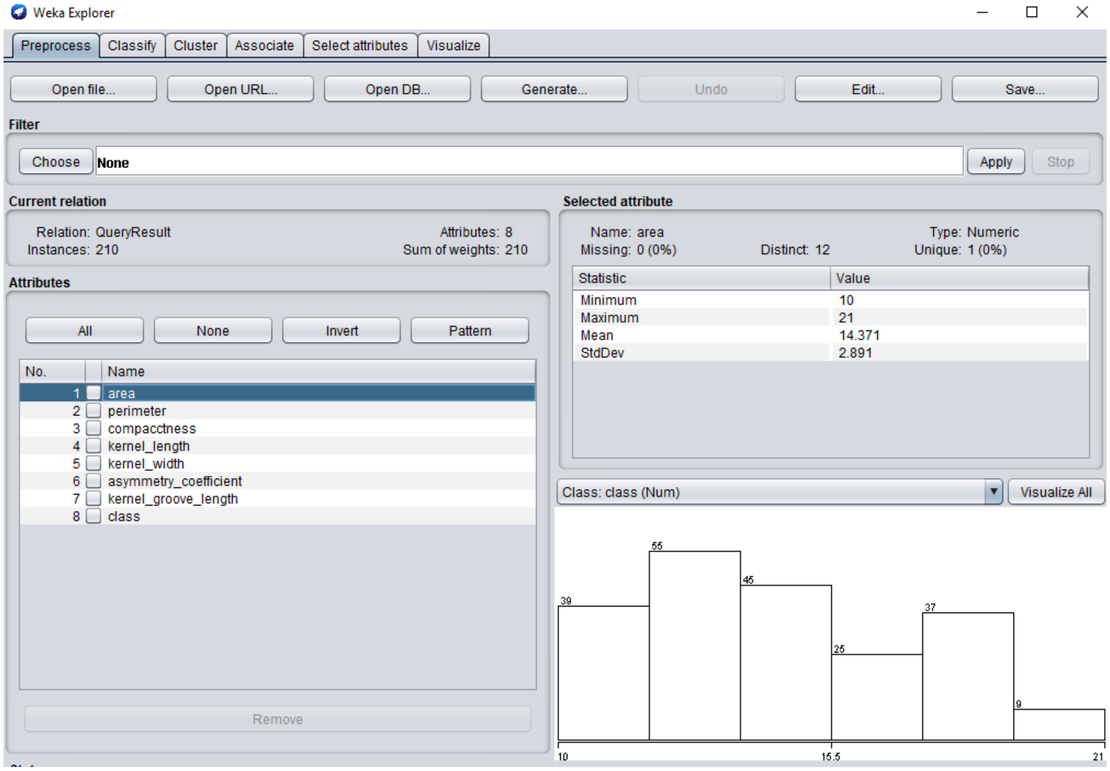
Relation: None
Instances: None

□ Generate sparse data

Current query: SELECT * FROM seeds_dataset_numeric

Then clicked OK. This will direct to the Weka Explorer.

Weka Explorer screen view is given below.



Filter

None

Current relation

Relation: QueryResult
Instances: 210
Attributes: 8
Sum of weights: 210

Selected attribute

Name: area
Missing: 0 (0%)
Distinct: 12
Type: Numeric
Unique: 1 (0%)

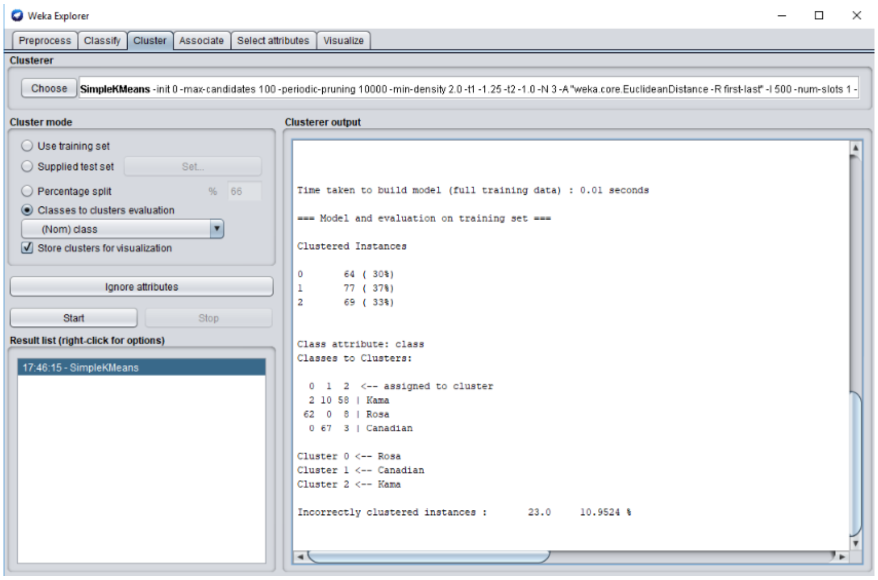
Statistic	Value
Minimum	10
Maximum	21
Mean	14.371
StdDev	2.891

Attributes

No.	Name
1	area
2	perimeter
3	compacctness
4	kernel_length
5	kernel_width
6	asymmetry_coefficient
7	kernel_groove_length
8	class

Class: class (Num)

All values including class comes in numeric format. Therefore, by using the weka filters by going to Choose->weka->filters->unsupervised->NumericToNominal I converted class attribute from numeric to nominal. Then I ran SimpleKMeans as same as done using the normal way which I did before I got the same inaccurate instances 14 and percentage as 10.9524%.



Clusterer

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 -

Cluster mode

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

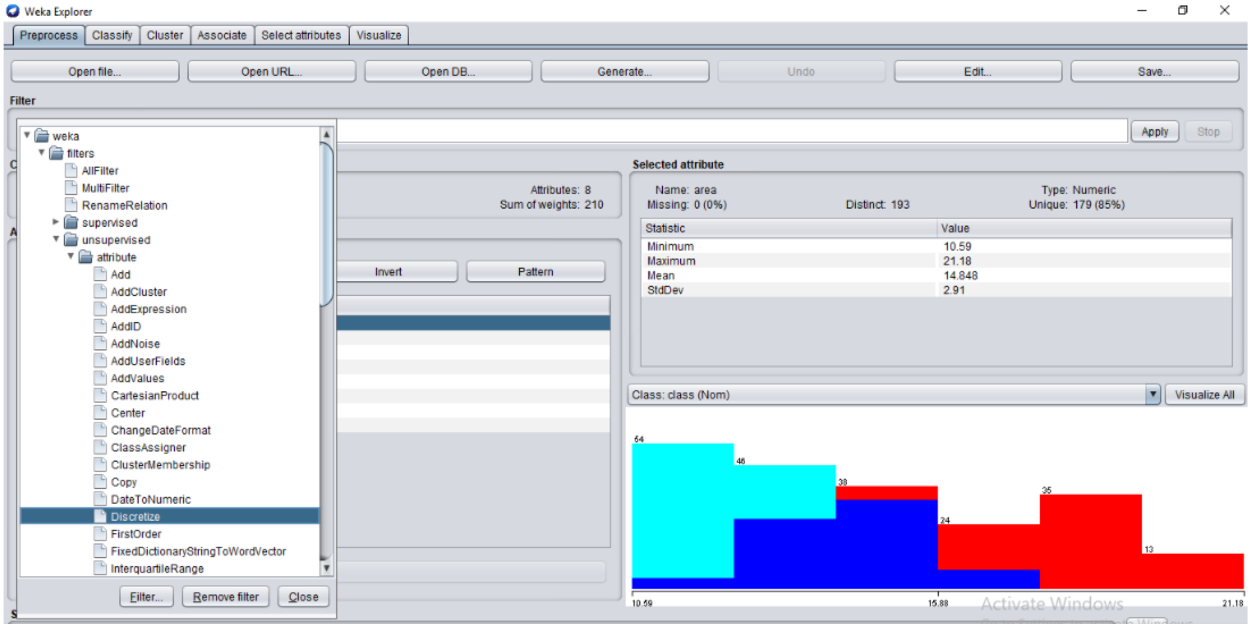
Result list (right-click for options)

17:46:15 - SimpleKMeans

Clusterer output

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

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

Clustered Instances

0       64 ( 30%)
1       77 ( 37%)
2       69 ( 33%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  2 10 58 | Kama
 62  0  8 | Rosa
  0 67  3 | Canadian

Cluster 0 <-- Rosa
Cluster 1 <-- Canadian
Cluster 2 <-- Kama

Incorrectly clustered instances :      23.0     10.9524 %
```

Then to make the dataset amenable for association rule mining all attributes should be converted to nominal. This was done by discretization as shown below.



Attributes: 8
Sum of weights: 210

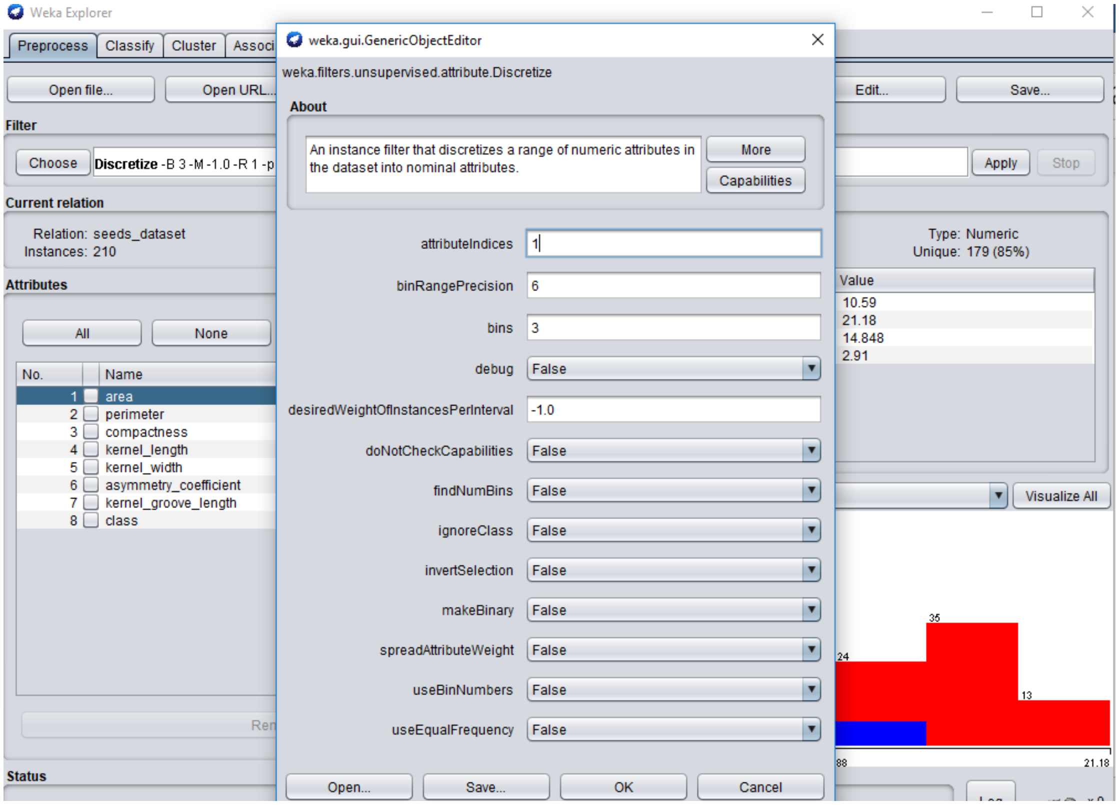
Selected attribute

Name: area
Missing: 0 (0%)
Distinct: 193
Type: Numeric
Unique: 179 (85%)

Statistic	Value
Minimum	10.59
Maximum	21.18
Mean	14.848
StdDev	2.91

Class: class (Nom)

I gave the number of bins as 3 and attributeIndices the attribute number (Parameter Setting)



Discretize -B 3 -M -1.0 -R 1 -p

Current relation

Relation: seeds_dataset
Instances: 210

Attributes

No.	Name
1	area
2	perimeter
3	compactness
4	kernel_length
5	kernel_width
6	asymmetry_coefficient
7	kernel_groove_length
8	class

Type: Numeric
Unique: 179 (85%)

Value
10.59
21.18
14.848
2.91

weka.gui.GenericObjectEditor

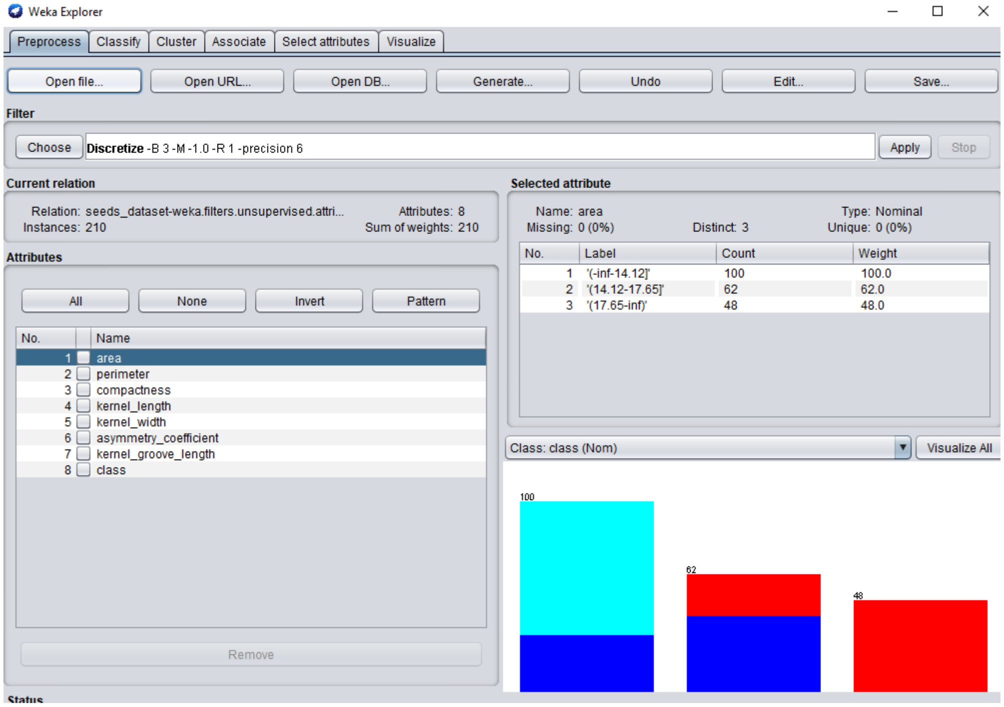
weka.filters.unsupervised.attribute.Discretize

About

An instance filter that discretizes a range of numeric attributes in the dataset into nominal attributes.

attributeIndices	1
binRangePrecision	6
bins	3
debug	False
desiredWeightOfInstancesPerInterval	-1.0
doNotCheckCapabilities	False
findNumBins	False
ignoreClass	False
invertSelection	False
makeBinary	False
spreadAttributeWeight	False
useBinNumbers	False
useEqualFrequency	False

Similarly, I gave number of bins as 3 after considering the amount of records in each attribute and got the output as below.



Filter: **Discretize** -B 3 -M -1.0 -R 1 -precision 6

Current relation

Relation: seeds_dataset-weka.filters.unsupervised.attri... Attributes: 8
Instances: 210 Sum of weights: 210

Attributes

No.	Name
1	□ area
2	□ perimeter
3	□ compactness
4	□ kernel_length
5	□ kernel_width
6	□ asymmetry_coefficient
7	□ kernel_groove_length
8	□ class

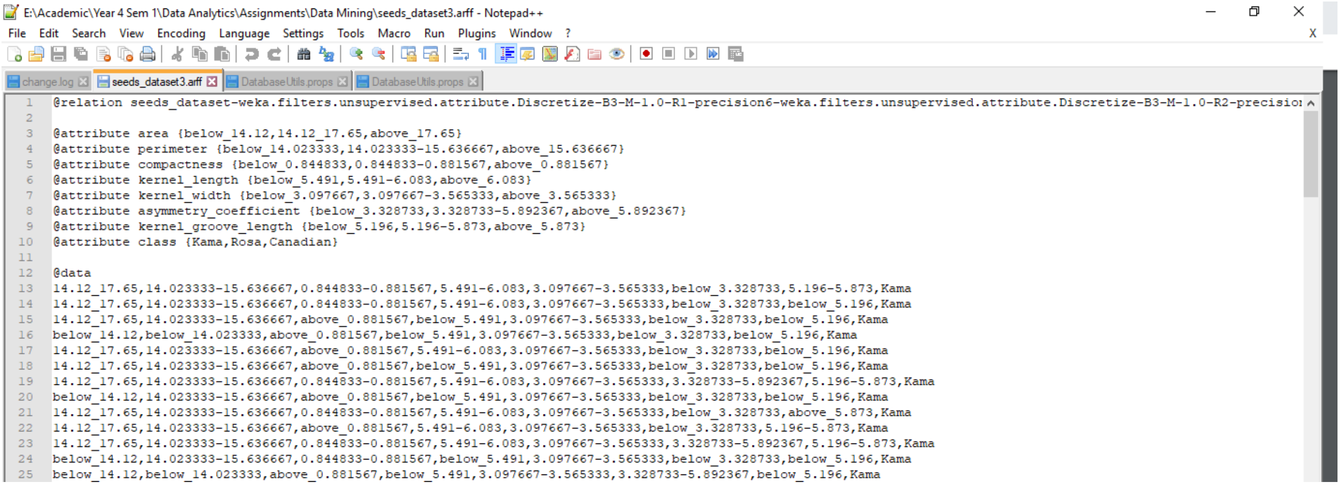
Selected attribute

Name: area Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count	Weight
1	'(-inf-14.12]'	100	100.0
2	'(14.12-17.65]'	62	62.0
3	'(17.65-inf)'	48	48.0

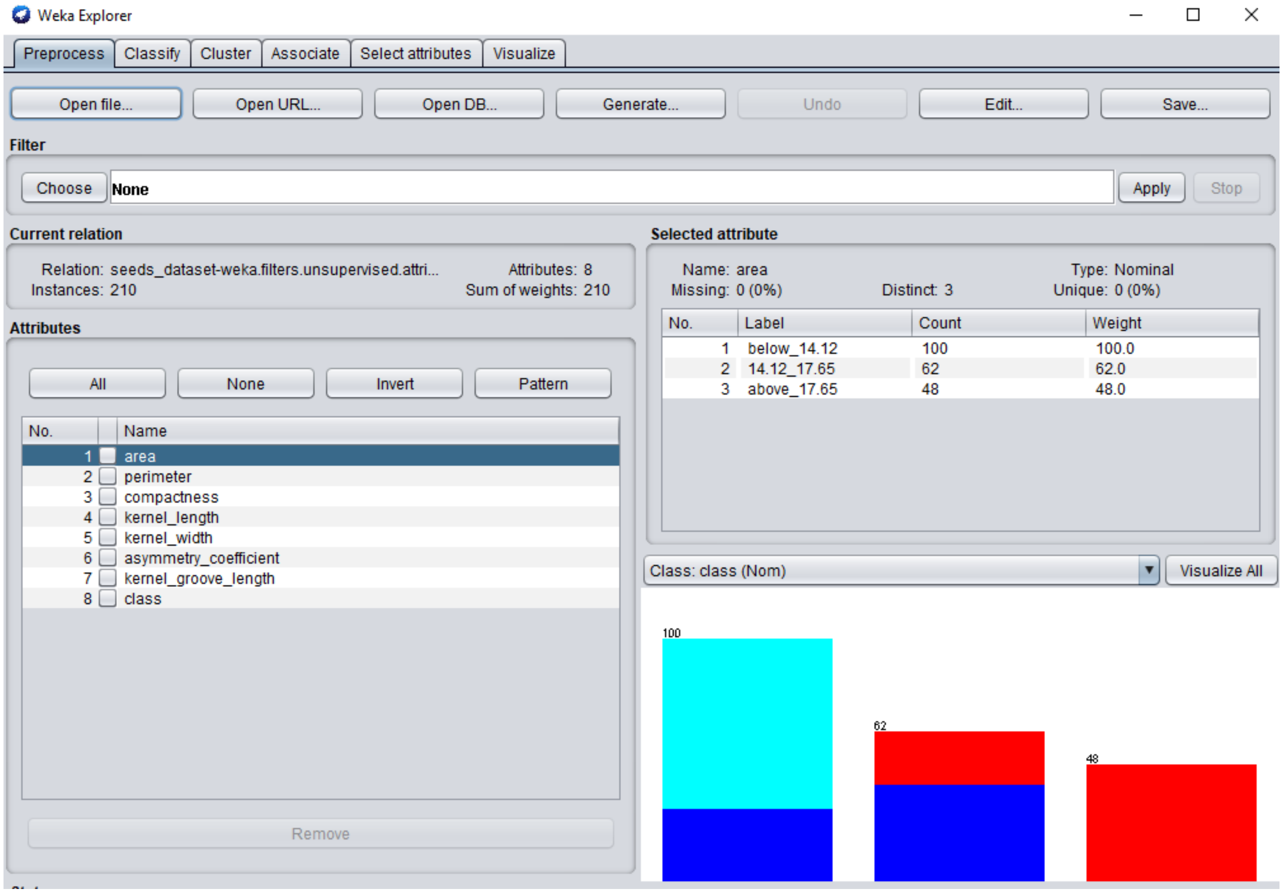
Class: class (Nom)

As labels are not meaningful, I had to edit using a text editor to make the values meaningful as shown below.



```
@relation seeds_dataset-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-precision6-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R2-precisio

@attribute area {below_14.12,14.12_17.65,above_17.65}
@attribute perimeter {below_14.023333,14.023333-15.636667,above_15.636667}
@attribute compactness {below_0.844833,0.844833-0.881567,above_0.881567}
@attribute kernel_length {below_5.491,5.491-6.083,above_6.083}
@attribute kernel_width {below_3.097667,3.097667-3.565333,above_3.565333}
@attribute asymmetry_coefficient {below_3.328733,3.328733-5.892367,above_5.892367}
@attribute kernel_groove_length {below_5.196,5.196-5.873,above_5.873}
@attribute class {Kama,Rosa,Canadian}

@data
14.12_17.65,14.023333-15.636667,0.844833-0.881567,5.491-6.083,3.097667-3.565333,below_3.328733,5.196-5.873,Kama
14.12_17.65,14.023333-15.636667,0.844833-0.881567,5.491-6.083,3.097667-3.565333,below_3.328733,below_5.196,Kama
14.12_17.65,14.023333-15.636667,above_0.881567,below_5.491,3.097667-3.565333,below_3.328733,below_5.196,Kama
below_14.12,below_14.023333,above_0.881567,below_5.491,3.097667-3.565333,below_3.328733,below_5.196,Kama
14.12_17.65,14.023333-15.636667,above_0.881567,5.491-6.083,3.097667-3.565333,below_3.328733,below_5.196,Kama
14.12_17.65,14.023333-15.636667,above_0.881567,below_5.491,3.097667-3.565333,below_3.328733,below_5.196,Kama
14.12_17.65,14.023333-15.636667,0.844833-0.881567,5.491-6.083,3.097667-3.565333,3.328733-5.892367,5.196-5.873,Kama
below_14.12,14.023333-15.636667,above_0.881567,below_5.491,3.097667-3.565333,below_3.328733,below_5.196,Kama
14.12_17.65,14.023333-15.636667,0.844833-0.881567,5.491-6.083,3.097667-3.565333,below_3.328733,above_5.873,Kama
14.12_17.65,14.023333-15.636667,above_0.881567,5.491-6.083,3.097667-3.565333,below_3.328733,5.196-5.873,Kama
14.12_17.65,14.023333-15.636667,0.844833-0.881567,5.491-6.083,3.097667-3.565333,3.328733-5.892367,5.196-5.873,Kama
below_14.12,14.023333-15.636667,0.844833-0.881567,below_5.491,3.097667-3.565333,below_3.328733,below_5.196,Kama
below_14.12,below_14.023333,above_0.881567,below_5.491,3.097667-3.565333,3.328733-5.892367,below_5.196,Kama
```

Then after opening saving it and opening it with weka the below window can be seen.



Filter

None

Current relation

Relation: seeds_dataset-weka.filters.unsupervised.attri... Attributes: 8
Instances: 210 Sum of weights: 210

Selected attribute

Name: area Type: Nominal
Missing: 0 (0%) Distinct: 3 Unique: 0 (0%)

No.	Label	Count	Weight
1	below_14.12	100	100.0
2	14.12_17.65	62	62.0
3	above_17.65	48	48.0

Attributes

No.		Name
1	□	area
2	□	perimeter
3	□	compactness
4	□	kernel_length
5	□	kernel_width
6	□	asymmetry_coefficient
7	□	kernel_groove_length
8	□	class

Class: class (Nom)

Now the data set is ready for the rule mining process.

Rule mining process

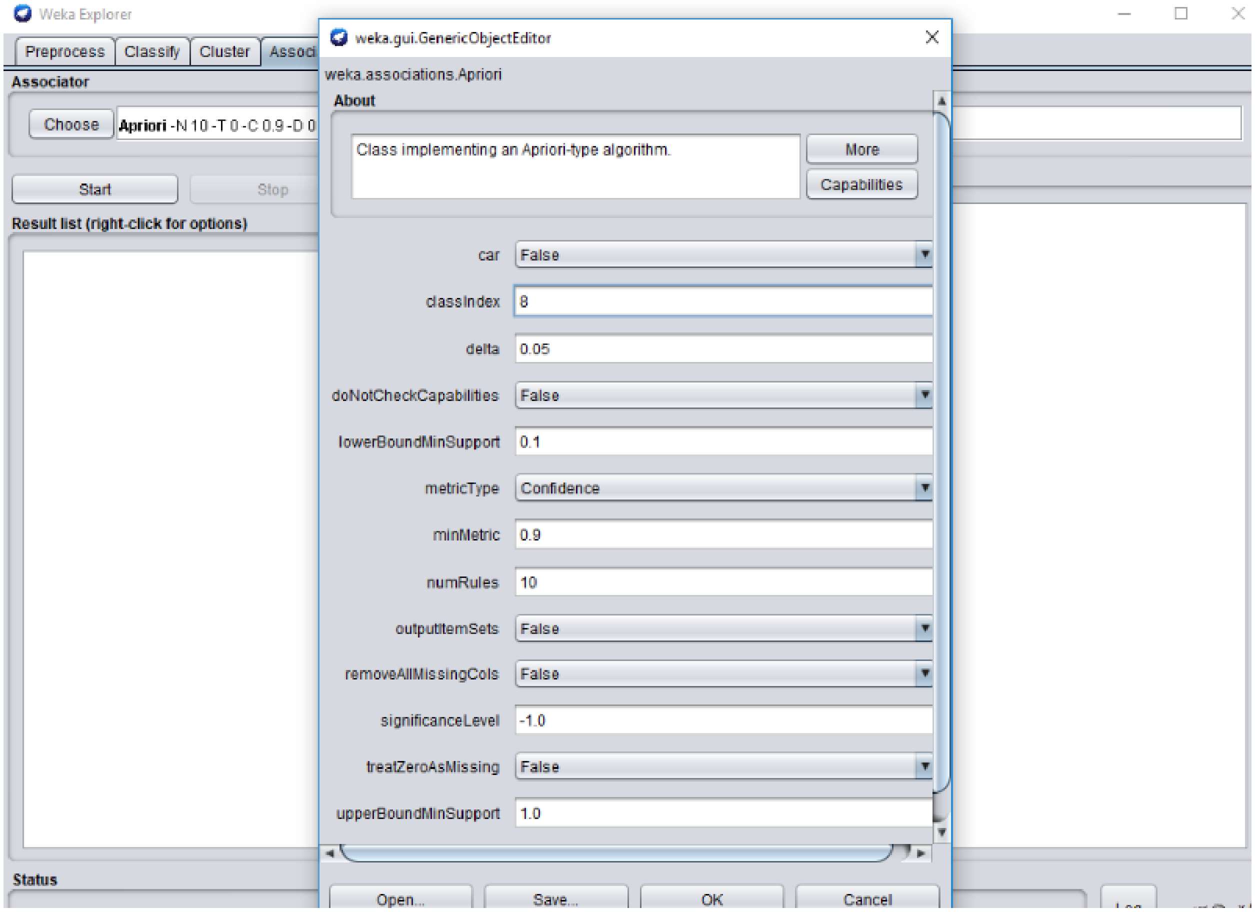
For the rule mining process, I selected the Apriori algorithm.

Reason for selecting Apriori algorithm

The Apriori Algorithm is considered an influential algorithm for mining frequent item sets for Boolean association rules. Apriori algorithm uses a "bottom up" approach, where frequent subsets are extended one item at a time (a step known as candidate generation) and then the groups of candidates are tested against the data.

The time required to run the Apriori algorithm after setting parameters for me was about 2 seconds. So Apriori Algorithm is fast.

Parameter Setting for the Apriori Algorithm was done as shown below.



Apriori -N 10 -T 0 -C 0.9 -D 0

weka.associations.Apriori

About

Class implementing an Apriori-type algorithm.

car	False
classIndex	8
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

I put the classindex as 8 because my 8th attribute is the class. I put lowerBoundMinSupport as 0.1 and upperBoundMinSupport, minMetric as 0.9 and firstly the number of rules were taken as 10 and the metricType as Confidence.

Later number of rules were taken as 1000 also to identify more rules.

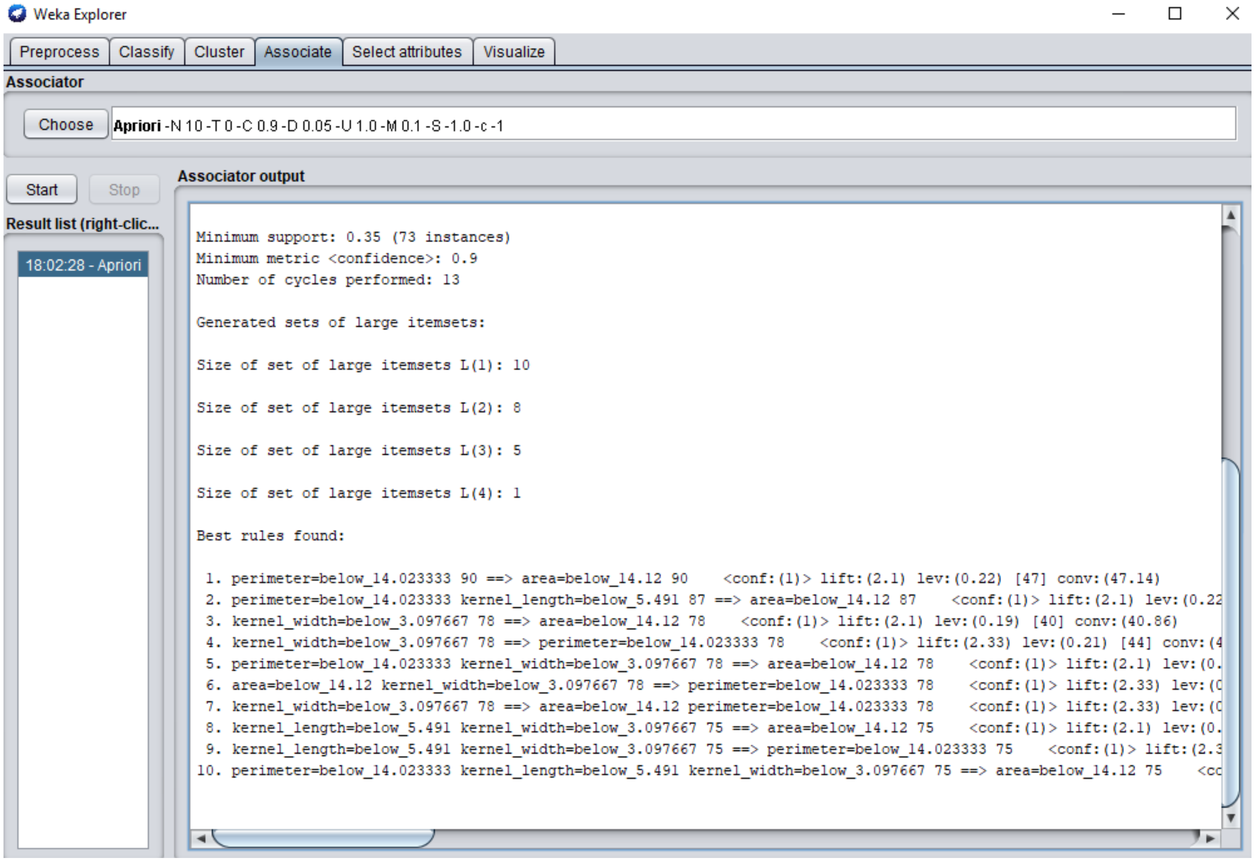
Similarly, I changed the car parameter to True to get a more accurate and direct relationship with the class attribute which contains the 3 wheat varieties Kama, Rosa and Canadian.

Lift metricType was not taken because when car is True no rules were found using the lift metricType.

Resulting rules

After running the Apriori algorithm the resulting rules for different parameter setting is given below.

1. Number of rules was taken as 10 and the metric type as confidence. Confidence level in all the 10 rules were 1 but since they are normally known they are not significant.13 cycles have been required to get the 10 rules where minMetric was 0.9.



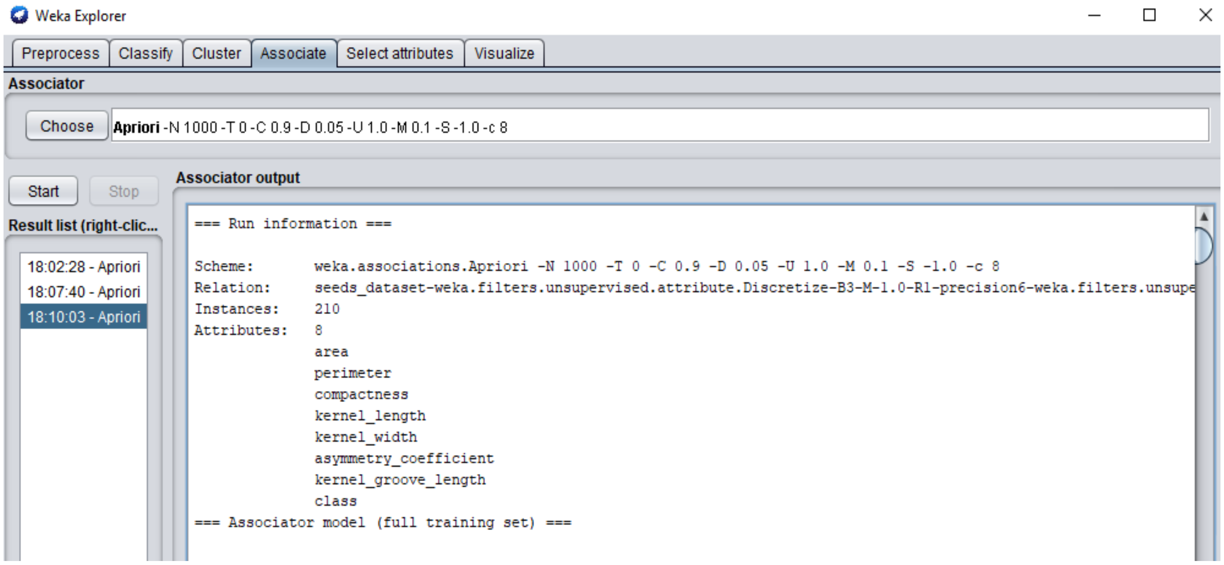
Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

```
Minimum support: 0.35 (73 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 13

Generated sets of large itemsets:

Size of set of large itemsets L(1): 10

Size of set of large itemsets L(2): 8

Size of set of large itemsets L(3): 5

Size of set of large itemsets L(4): 1

Best rules found:

 1. perimeter=below_14.023333 90 ==> area=below_14.12 90    <conf:(1)> lift:(2.1) lev:(0.22) [47] conv:(47.14)
 2. perimeter=below_14.023333 kernel_length=below_5.491 87 ==> area=below_14.12 87    <conf:(1)> lift:(2.1) lev:(0.22
 3. kernel_width=below_3.097667 78 ==> area=below_14.12 78    <conf:(1)> lift:(2.1) lev:(0.19) [40] conv:(40.86)
 4. kernel_width=below_3.097667 78 ==> perimeter=below_14.023333 78    <conf:(1)> lift:(2.33) lev:(0.21) [44] conv:(4
 5. perimeter=below_14.023333 kernel_width=below_3.097667 78 ==> area=below_14.12 78    <conf:(1)> lift:(2.1) lev:(0.
 6. area=below_14.12 kernel_width=below_3.097667 78 ==> perimeter=below_14.023333 78    <conf:(1)> lift:(2.33) lev:(0
 7. kernel_width=below_3.097667 78 ==> area=below_14.12 perimeter=below_14.023333 78    <conf:(1)> lift:(2.33) lev:(0
 8. kernel_length=below_5.491 kernel_width=below_3.097667 75 ==> area=below_14.12 75    <conf:(1)> lift:(2.1) lev:(0.
 9. kernel_length=below_5.491 kernel_width=below_3.097667 75 ==> perimeter=below_14.023333 75    <conf:(1)> lift:(2.3
10. perimeter=below_14.023333 kernel_length=below_5.491 kernel_width=below_3.097667 75 ==> area=below_14.12 75    <c
```

(Refer ‘Confidence,rules=10.txt’ for the rules)

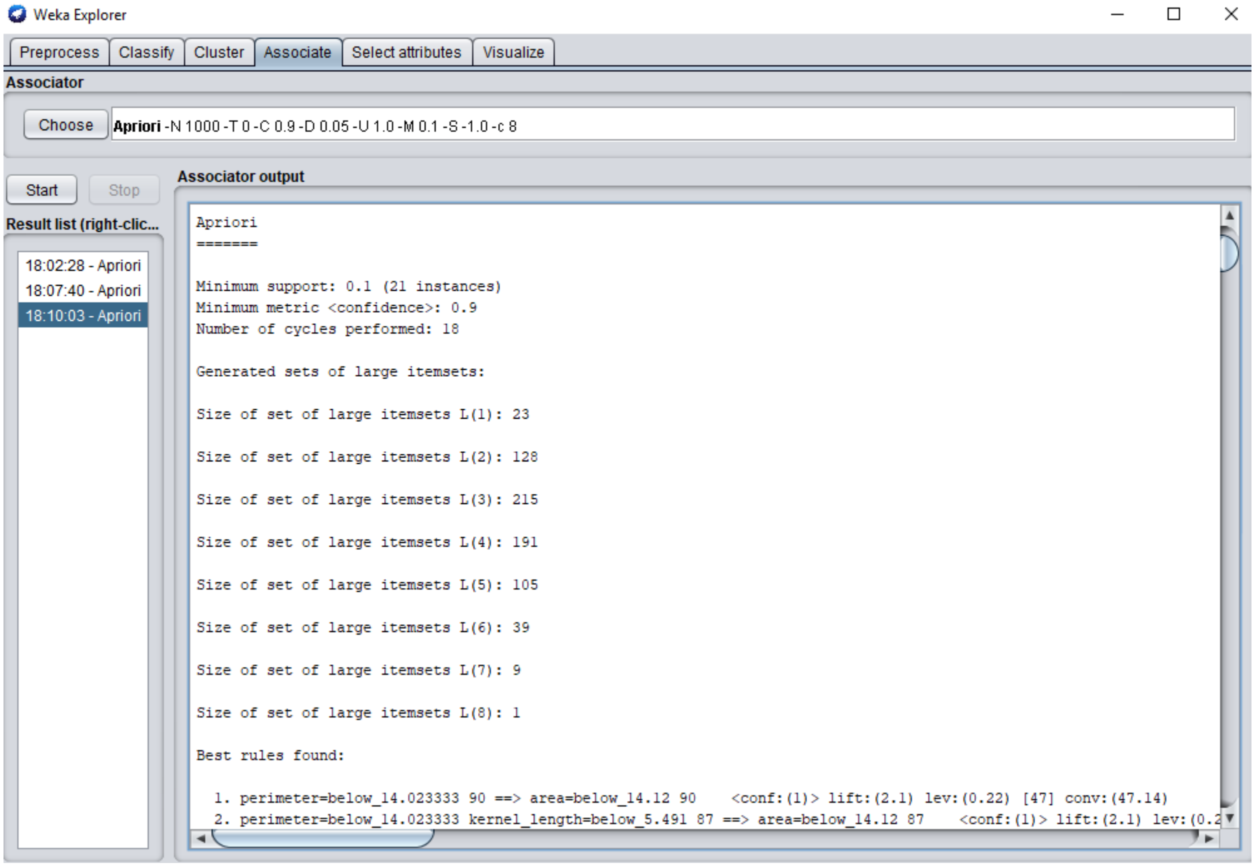
Confidence level in the rules varies from 0 to 1 and 1 is then taken as the best confidence.

2. Number of rules was taken as 1000 and the metric type as confidence.18 cycles have been required to get the 1000 rules where the minMetric was 0.9.



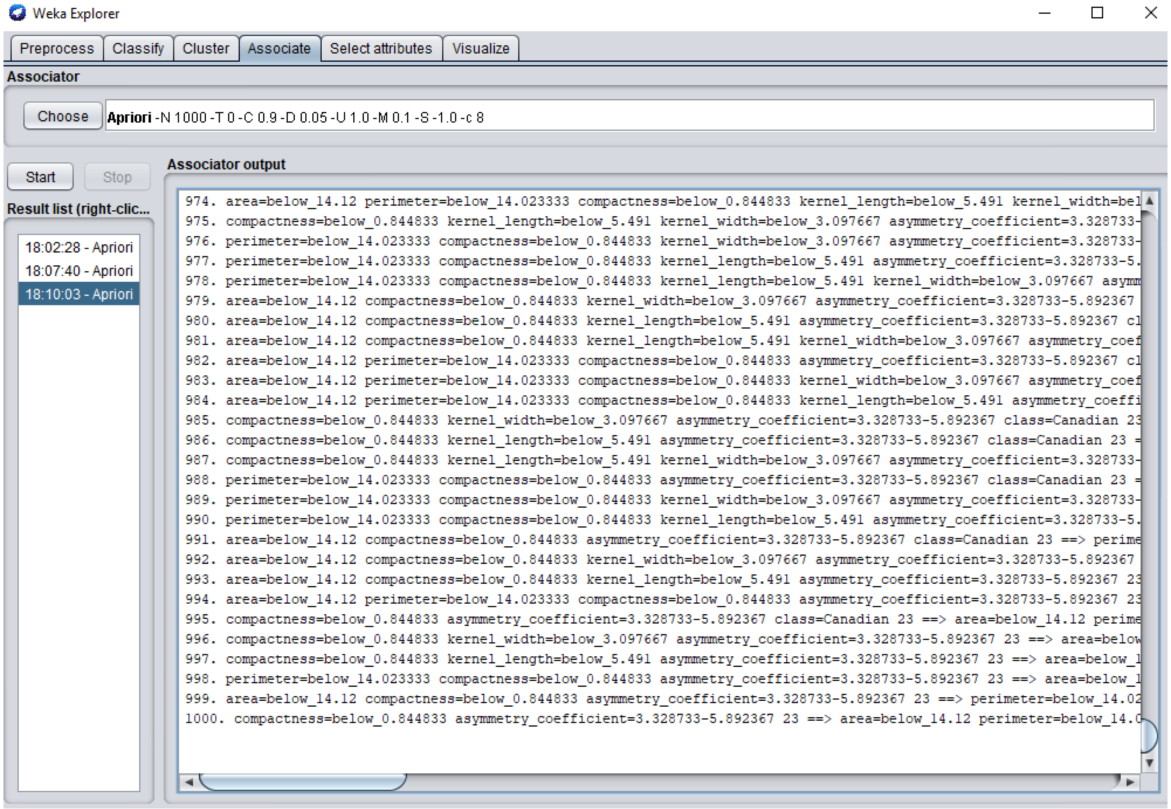
Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 8

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 8
Relation:     seeds_dataset-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-precision6-weka.filters.unsupe
Instances:    210
Attributes:   8
              area
              perimeter
              compactness
              kernel_length
              kernel_width
              asymmetry_coefficient
              kernel_groove_length
              class
=== Associator model (full training set) ===
```



Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 8

```
Apriori
=======

Minimum support: 0.1 (21 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 23

Size of set of large itemsets L(2): 128

Size of set of large itemsets L(3): 215

Size of set of large itemsets L(4): 191

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

Size of set of large itemsets L(6): 39

Size of set of large itemsets L(7): 9

Size of set of large itemsets L(8): 1

Best rules found:

  1. perimeter=below_14.023333 90 ==> area=below_14.12 90    <conf:(1)> lift:(2.1) lev:(0.22) [47] conv:(47.14)
  2. perimeter=below_14.023333 kernel_length=below_5.491 87 ==> area=below_14.12 87    <conf:(1)> lift:(2.1) lev:(0.2
```



Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c 8

```
974. area=below_14.12 perimeter=below_14.023333 compactness=below_0.844833 kernel_length=below_5.491 kernel_width=bel
975. compactness=below_0.844833 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-
976. perimeter=below_14.023333 compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-
977. perimeter=below_14.023333 compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.
978. perimeter=below_14.023333 compactness=below_0.844833 kernel_length=below_5.491 kernel_width=below_3.097667 asymm
979. area=below_14.12 compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367
980. area=below_14.12 compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.892367 cl
981. area=below_14.12 compactness=below_0.844833 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coef
982. area=below_14.12 perimeter=below_14.023333 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 cl
983. area=below_14.12 perimeter=below_14.023333 compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coef
984. area=below_14.12 perimeter=below_14.023333 compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coeffi
985. compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 class=Canadian 23
986. compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.892367 class=Canadian 23 =
987. compactness=below_0.844833 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-
988. perimeter=below_14.023333 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 class=Canadian 23 =
989. perimeter=below_14.023333 compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-
990. perimeter=below_14.023333 compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.
991. area=below_14.12 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 class=Canadian 23 ==> perime
992. area=below_14.12 compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367
993. area=below_14.12 compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.892367 23
994. area=below_14.12 perimeter=below_14.023333 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 23
995. compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 class=Canadian 23 ==> area=below_14.12 perime
996. compactness=below_0.844833 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 23 ==> area=below
997. compactness=below_0.844833 kernel_length=below_5.491 asymmetry_coefficient=3.328733-5.892367 23 ==> area=below_l
998. perimeter=below_14.023333 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 23 ==> area=below_l
999. area=below_14.12 compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 23 ==> perimeter=below_14.02
1000. compactness=below_0.844833 asymmetry_coefficient=3.328733-5.892367 23 ==> area=below_14.12 perimeter=below_14.0
```

(Refer ‘Confidence,rules=1000.txt’ for the rules)

Some interesting rule descriptions found are

i) If perimeter is above 15.636667 then (==>) the wheat variety would be Rosa (class=Rosa). This was shown with a confidence of 1.

ii) If perimeter is above 15.636667 and kernel width is above 3.565333 48 then (==>) the wheat variety would be Rosa (class=Rosa). This was shown with a confidence of 1.

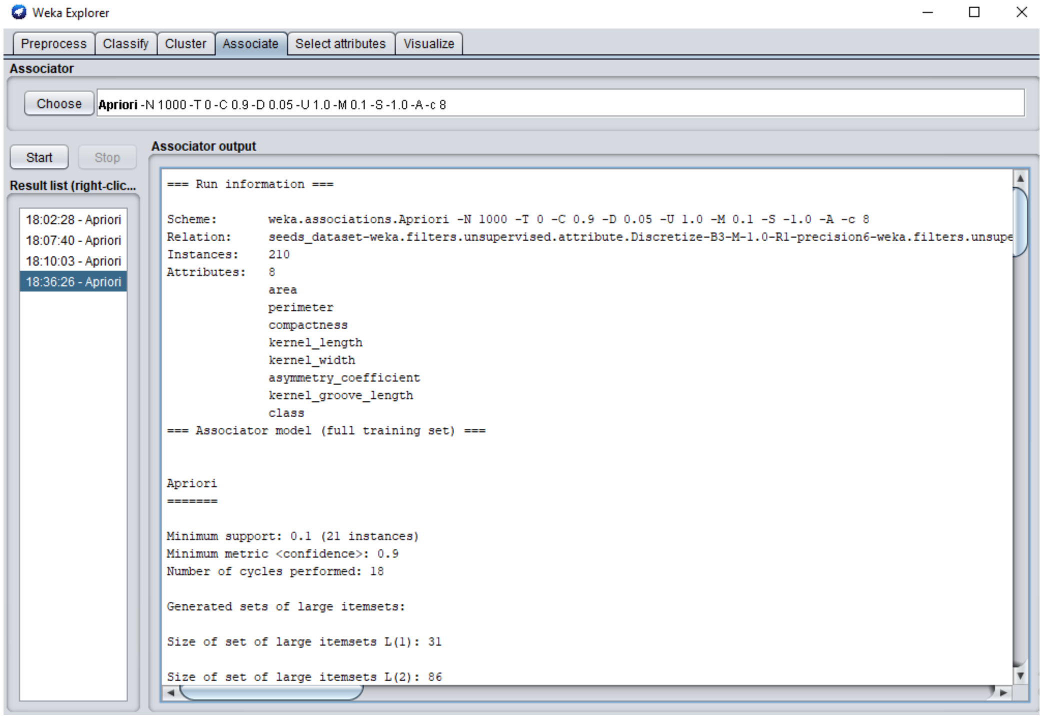
iii) If perimeter is between 14.023333 and 15.636667and asymmetry coefficient is below3.328733 and kernel groove length is below_5.196 then (==>) kernel width is between 3.097667 and 3.565333 and the wheat variety would be Kama (class=Kama). This was shown with a confidence of 1.

iv) If area is below 14.12 and perimeter is below14.023333 and compactness is below 0.844833 and kernel length is below 5.491 and kernel width is below 3.097667 and asymmetry coefficient is between 3.328733 and 5.892367 then (==>) wheat variety is Canadian (class=Canadian). This was shown with a confidence of 1.

(All the attributes are the geometrical properties of kernels belonging to the three different varieties of wheat namely Kama, Rosa and Canadian)

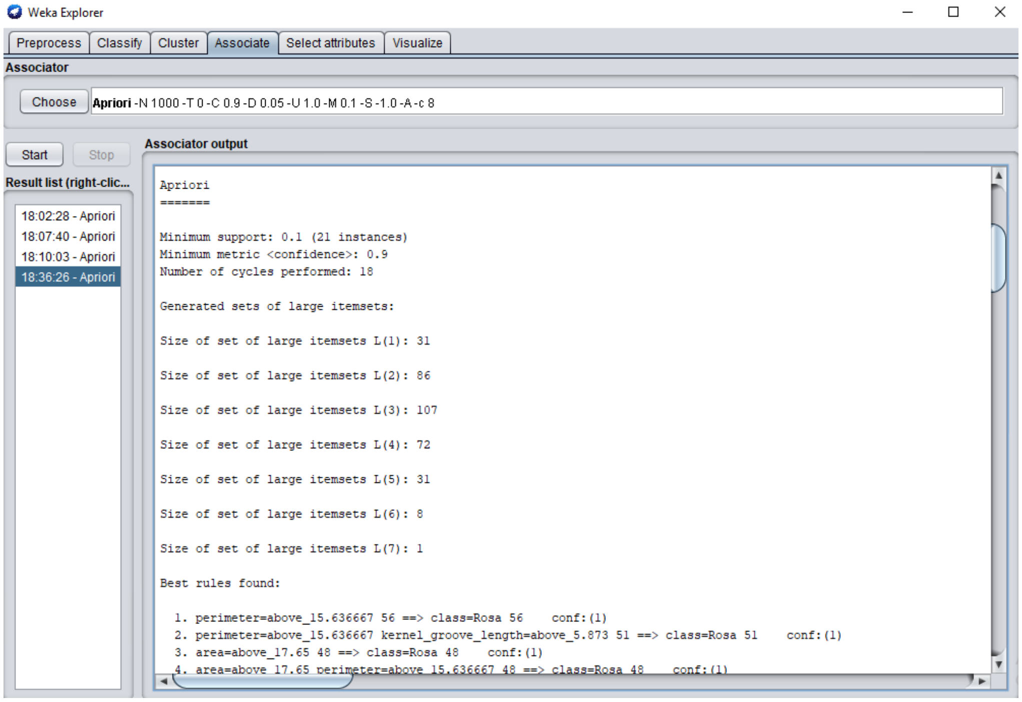
All the above were done taking the car as False. More direct relationships can be taken by taking car as True so that the right side will have the class attribute.

3. When the car parameter is taken as true and the number of rules is given as 1000 and the metricType as confidence we can see 155 rules as seen below. The number of cycles is 18 and the minMetric is 0.9.



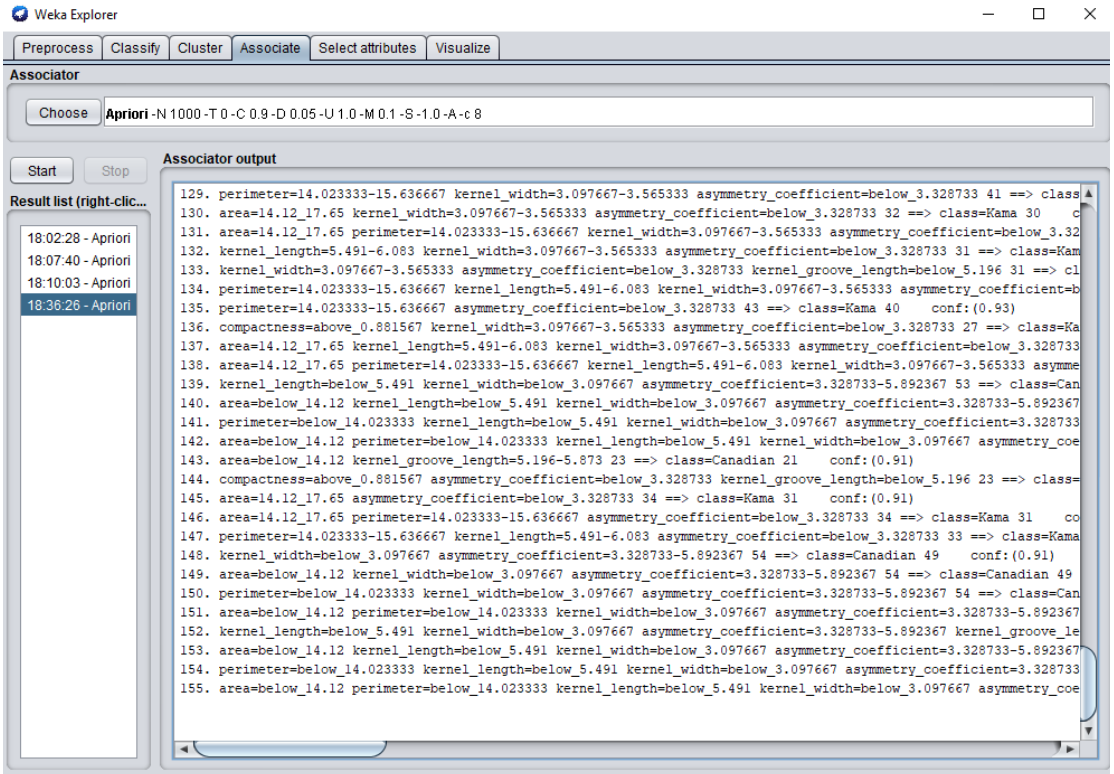
Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c 8

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c 8
Relation:     seeds_dataset-weka.filters.unsupervised.attribute.Discretize-B3-M-1.0-R1-precision6-weka.filters.unsupe
Instances:    210
Attributes:   8
              area
              perimeter
              compactness
              kernel_length
              kernel_width
              asymmetry_coefficient
              kernel_groove_length
              class
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.1 (21 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 31

Size of set of large itemsets L(2): 86
```



Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c 8

```
Apriori
=======

Minimum support: 0.1 (21 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 18

Generated sets of large itemsets:

Size of set of large itemsets L(1): 31

Size of set of large itemsets L(2): 86

Size of set of large itemsets L(3): 107

Size of set of large itemsets L(4): 72

Size of set of large itemsets L(5): 31

Size of set of large itemsets L(6): 8

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

Best rules found:

  1. perimeter=above_15.636667 56 ==> class=Rosa 56    conf:(1)
  2. perimeter=above_15.636667 kernel_groove_length=above_5.873 51 ==> class=Rosa 51    conf:(1)
  3. area=above_17.65 48 ==> class=Rosa 48    conf:(1)
  4. area=above_17.65 perimeter=above_15.636667 48 ==> class=Rosa 48    conf:(1)
```



Apriori -N 1000 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -A -c 8

```
129. perimeter=14.023333-15.636667 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733 41 ==> class
130. area=14.12_17.65 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733 32 ==> class=Kama 30    c
131. area=14.12_17.65 perimeter=14.023333-15.636667 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.32
132. kernel_length=5.491-6.083 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733 31 ==> class=Kam
133. kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733 kernel_groove_length=below_5.196 31 ==> cl
134. perimeter=14.023333-15.636667 kernel_length=5.491-6.083 kernel_width=3.097667-3.565333 asymmetry_coefficient=b
135. perimeter=14.023333-15.636667 asymmetry_coefficient=below_3.328733 43 ==> class=Kama 40    conf:(0.93)
136. compactness=above_0.881567 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733 27 ==> class=Ka
137. area=14.12_17.65 kernel_length=5.491-6.083 kernel_width=3.097667-3.565333 asymmetry_coefficient=below_3.328733
138. area=14.12_17.65 perimeter=14.023333-15.636667 kernel_length=5.491-6.083 kernel_width=3.097667-3.565333 asymme
139. kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 53 ==> class=Can
140. area=below_14.12 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367
141. perimeter=below_14.023333 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733
142. area=below_14.12 perimeter=below_14.023333 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coe
143. area=below_14.12 kernel_groove_length=5.196-5.873 23 ==> class=Canadian 21    conf:(0.91)
144. compactness=above_0.881567 asymmetry_coefficient=below_3.328733 kernel_groove_length=below_5.196 23 ==> class=
145. area=14.12_17.65 asymmetry_coefficient=below_3.328733 34 ==> class=Kama 31    conf:(0.91)
146. area=14.12_17.65 perimeter=14.023333-15.636667 asymmetry_coefficient=below_3.328733 34 ==> class=Kama 31    co
147. perimeter=14.023333-15.636667 kernel_length=5.491-6.083 asymmetry_coefficient=below_3.328733 33 ==> class=Kama
148. kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 54 ==> class=Canadian 49    conf:(0.91)
149. area=below_14.12 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 54 ==> class=Canadian 49
150. perimeter=below_14.023333 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 54 ==> class=Can
151. area=below_14.12 perimeter=below_14.023333 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367
152. kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367 kernel_groove_le
153. area=below_14.12 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733-5.892367
154. perimeter=below_14.023333 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coefficient=3.328733
155. area=below_14.12 perimeter=below_14.023333 kernel_length=below_5.491 kernel_width=below_3.097667 asymmetry_coe
```

All these 155 rules are very interesing rules and this 155 rules will be the selection which would be shown to the client. From the 155 rules the top or the first 104 rules are the best rules with confidence 1. So the best selection to show for the client is this 104 rules. The 155 rules is attached with this report which is in a text file. (Refer 'Confidence,Car=true.txt' for the rules)

Some one to one or simple relationship rule descriptions from the 155 rules are given below.

i. If perimeter is above 15.636667 then (==>) the wheat variety would be Rosa (class=Rosa). This was shown with a confidence of 1.
ii. If area is above 17.65 then(==>) the wheat variety would be Rosa (class=Rosa). This was shown with a confidence of 1.
iii. If compactness is below 0.844833 then (==>) the wheat variety would be Canadian (class=Canadian). This was shown with a confidence of 0.97.
iv. If area is 14.12_17.65 and asymmetry coefficient is below 3.328733 then (==>) the wheat variety would be Kama (class=Kama). This was shown with a confidence of 0.91.
v. If area is below 14.12 and kernel groove length is between 5.196 and 5.873 then (==>) the wheat variety would be Canadian (class=Canadian). This was shown with a confidence of 0.91.

(All the attributes are the geometrical properties of kernels belonging to the three different varieties of wheat namely Kama, Rosa and Canadian)

Recommendations

Mainly to identify the 3 wheat varieties Kama, Rosa and Canadian I can recommend 3 recommendations considering each wheat variety to the client.

1) If perimeter of the kernel is greater than 15.64 and area of the kernel is greater than 17.65 then I would recommend considering the wheat variety as Rosa.
2) If kernel width is below 3.10 and kernel groove length is between 5.20 and 5.87, I would recommend considering the wheat variety as Canadian.
3) If perimeter is between 14.02 and 15.64 and asymmetry coefficient is below 3.33 and kernel groove length is below 5.20, I would be recommend considering the wheat variety as Canadian or perimeter is between 14.02 and 15.64 and kernel width is between 3.10 and 3.57 and kernel groove length is below 5.20, I would recommend considering the wheat variety as Canadian.

(All values are taken after rounding to the second decimal place)

Above 3 recommendations were taken after considering the rules with confidence equal to 1.

From above recommendations the client can simple identify the 3 wheat varieties Kama, Rosa and Canadian separately.