SCS 4204/ IS 4103/ CS 4104 - Data Analytics
Assignment 01
Index Number - 19001576

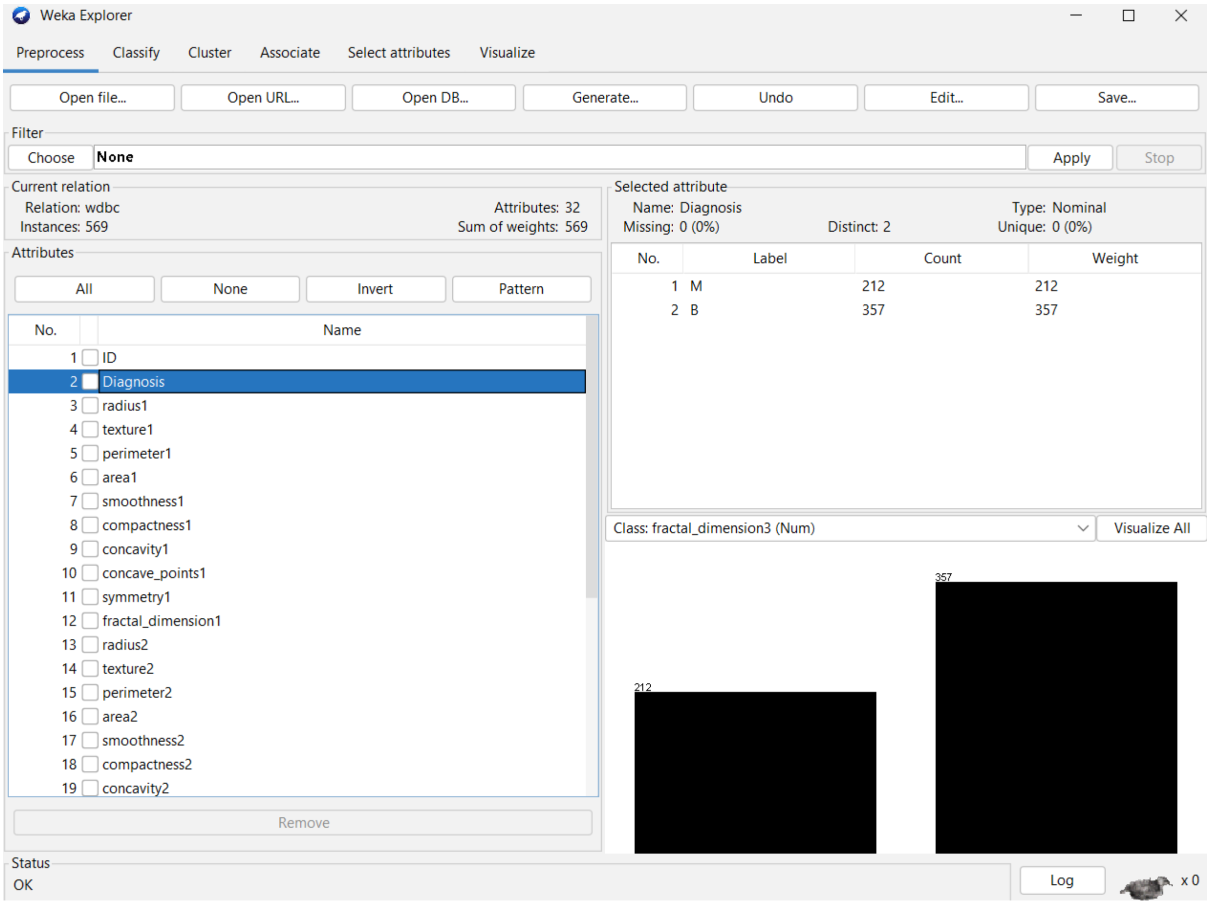
Task 01

Dataset name : Breast Cancer Wisconsin (Diagnostic)

Description: Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

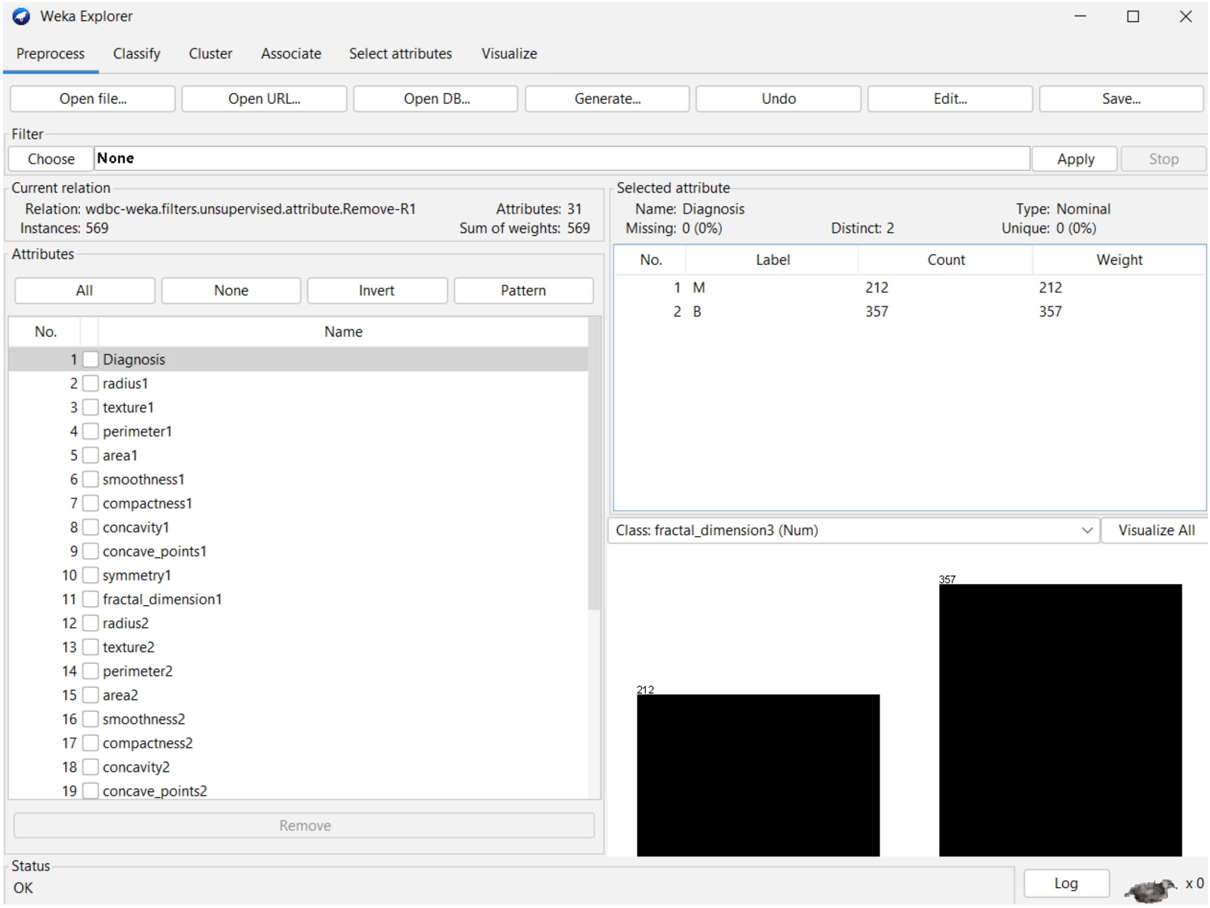
Attributes: 30
Target: Diagnosis
Values: (M = malignant, B = benign)

<u>Load dataset</u>



Preprocessing

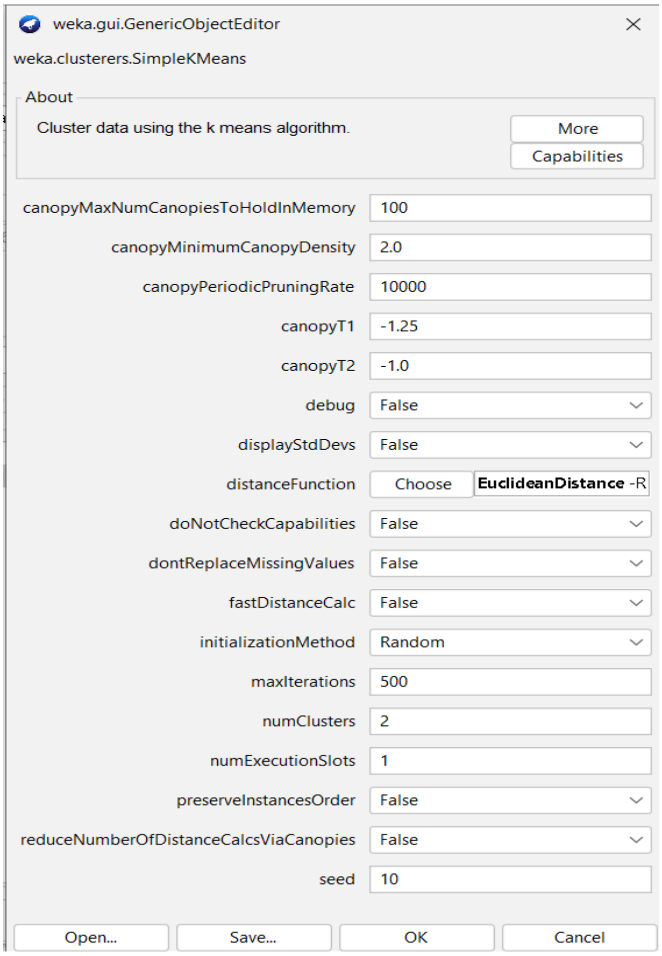
Removing ID attribute



<u>Clustering using K-Means</u>

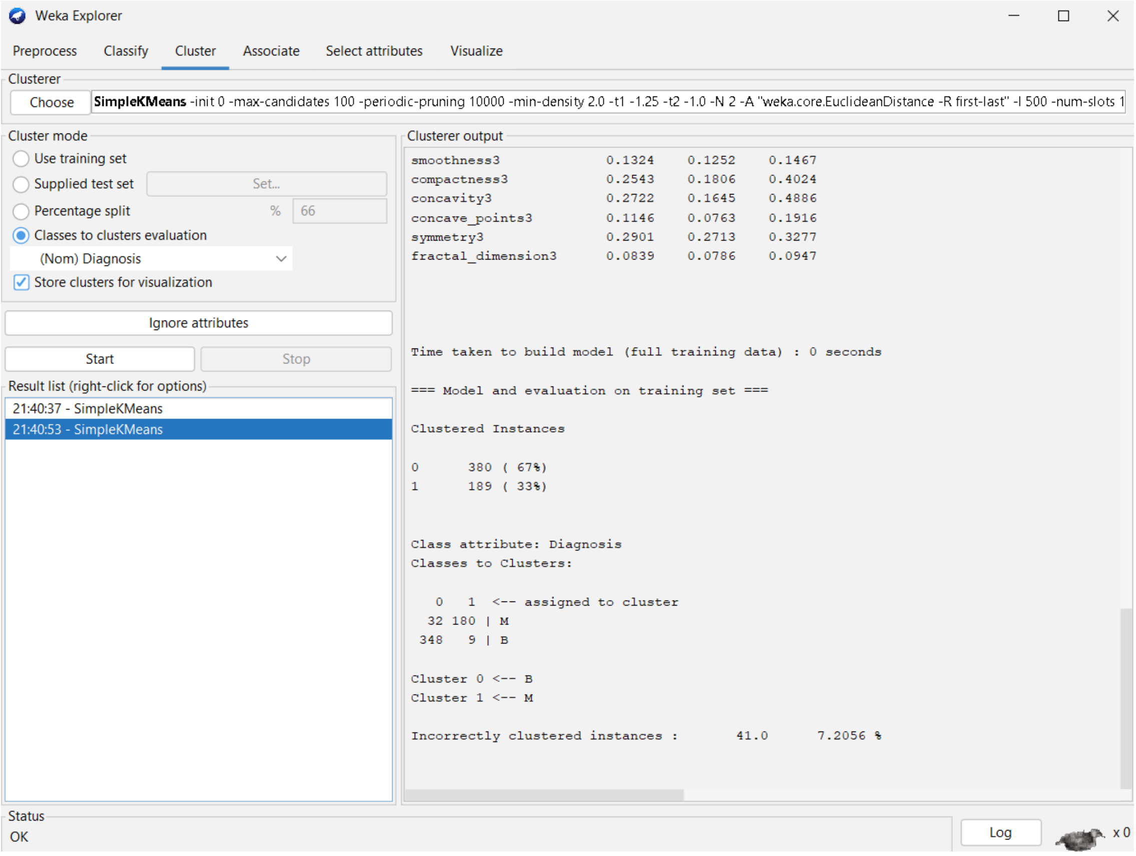
For KMeans, Default parameter values gave higher accuracy.

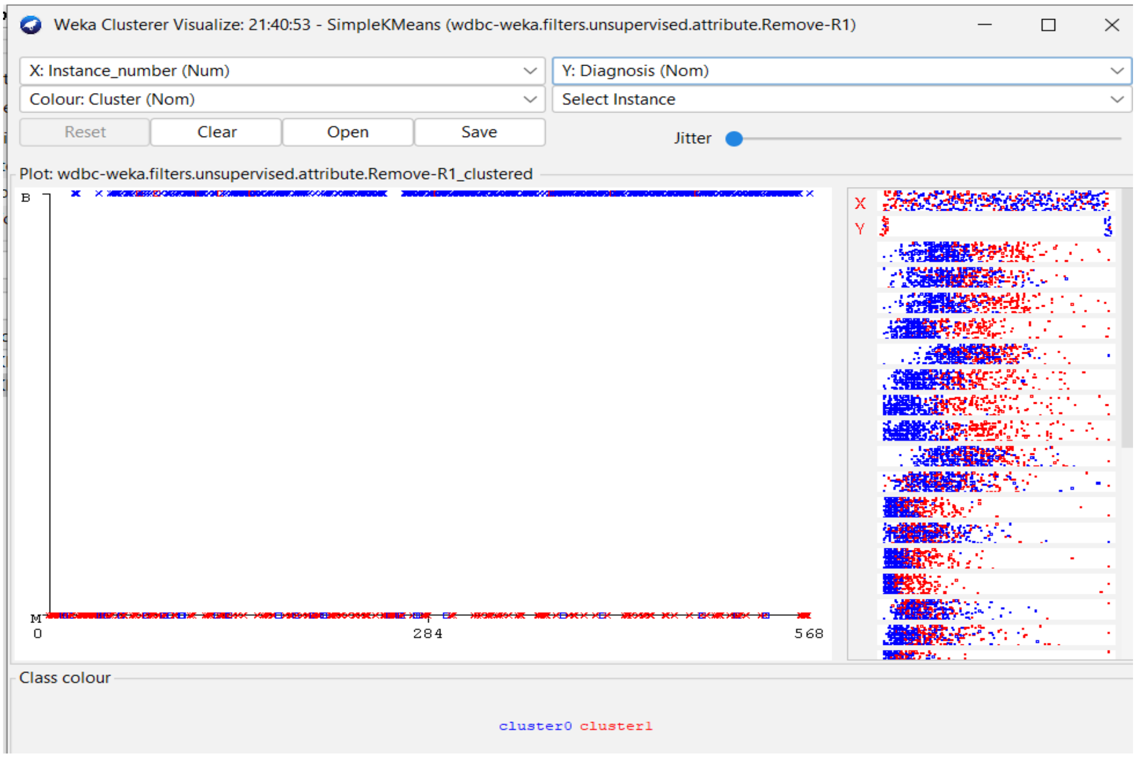
Parameter settings



Output

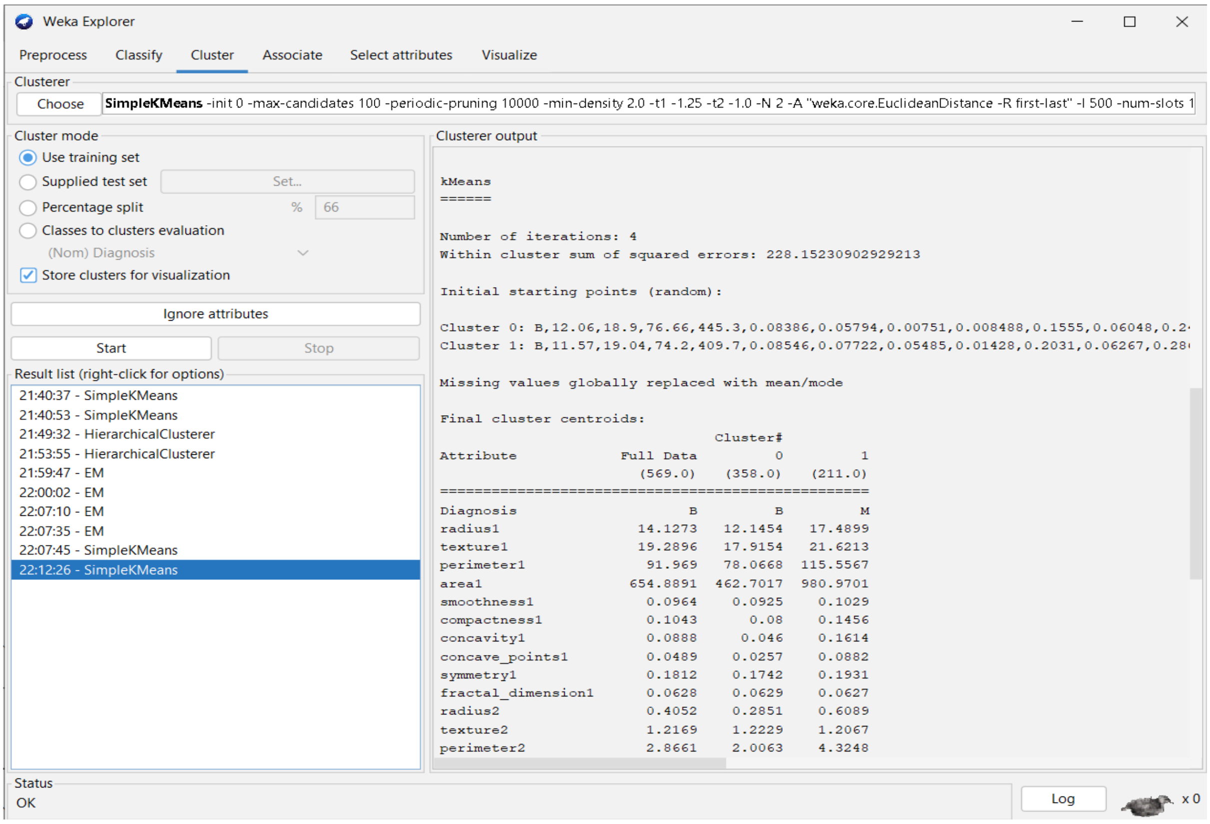
Cluster mode: Classes to cluster evaluation

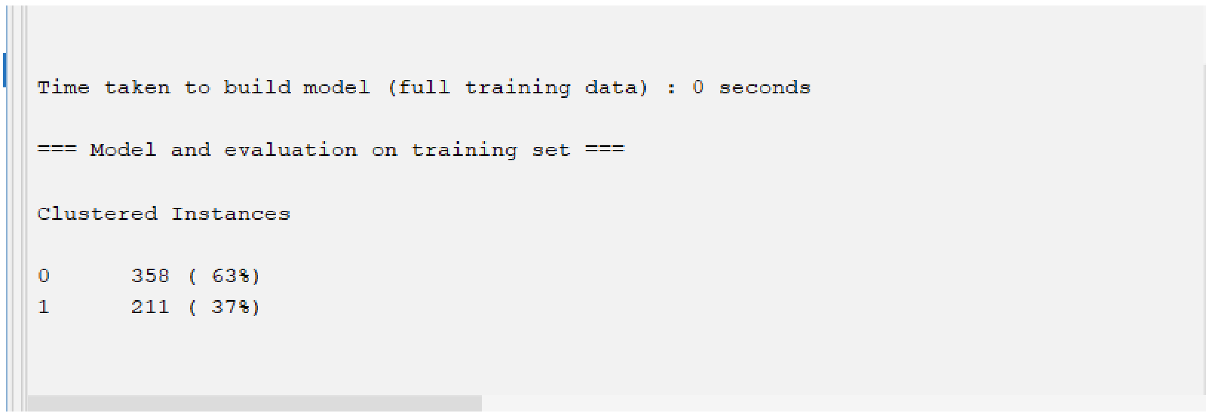




Accuracy (Correctly clustered instances) : (100 - 7.2056) = 92.7944 %

Cluster mode: Use training set





Number of iterations : 4
Sum of squared errors : 228.1523
Time taken : 0 seconds

Analysis of output

Clusterer output

Attribute	Full Data (569.0)	0 (358.0)	1 (211.0)
Diagnosis	B	B	M
radius1	14.1273	12.1454	17.4899
texture1	19.2896	17.9154	21.6213
perimeter1	91.969	78.0668	115.5567
area1	654.8891	462.7017	980.9701
smoothness1	0.0964	0.0925	0.1029
compactness1	0.1043	0.08	0.1456
concavity1	0.0888	0.046	0.1614
concave_points1	0.0489	0.0257	0.0882
symmetry1	0.1812	0.1742	0.1931
fractal_dimension1	0.0628	0.0629	0.0627
radius2	0.4052	0.2851	0.6089
texture2	1.2169	1.2229	1.2067
perimeter2	2.8661	2.0063	4.3248
area2	40.3371	21.2133	72.7841
smoothness2	0.007	0.0072	0.0068
compactness2	0.0255	0.0214	0.0324
concavity2	0.0319	0.026	0.042
concave_points2	0.0118	0.0099	0.0151
symmetry2	0.0205	0.0206	0.0205
fractal_dimension2	0.0038	0.0036	0.0041
radius3	16.2692	13.3797	21.1717
texture3	25.6772	23.5147	29.3463
perimeter3	107.2612	87.0006	141.637
area3	880.5831	558.8846	1426.4033
smoothness3	0.1324	0.1249	0.145
compactness3	0.2543	0.1824	0.3762
concavity3	0.2722	0.1659	0.4525
concave_points3	0.1146	0.0744	0.1828
symmetry3	0.2901	0.27	0.3241
fractal_dimension3	0.0839	0.0794	0.0916

The output shows some hidden insights that can be useful. When selecting the number of clusters 2 the sum of squared error within the cluster is 228.1523. One cluster has B as diagnosis and another cluster has M as diagnosis. If we analyse the centroid of each attribute value, radius1 has centroid value 14.1273 for full data, 12.1454 for cluster with diagnosis B and 17.4899 for cluster with diagnose M. So we can make the condition that if radius1 is less than 12.1454 then there is a high possibility that diagnosis is B, if the radius is within the range from 12.1454 to 14.1273, then also there is a possibility that the corresponding diagnosis will be B.

if radius1 is greater than 17.4899 then there is a high possibility that diagnosis is M, if the radius is within the range from 14.1273 to 17.4899, then also there is a possibility that the corresponding diagnosis will be M. Like this we can make conditions for each attribute value by analysing its centroids values. If we check this condition with the dataset this satisfies most of the instances in the dataset.

When a new data is given as input, these conditions can be checked and take the most optimal output as a decision that satisfies many numbers of created conditions.

Some other conditions

If area1 < 462.7017 , then diagnosis = B
If 462.7017 <= area1 < 654.8891 , then diagnosis = B
If 980.9701 <= area1 , then diagnosis = M
If 654.8891 <= area1 < 980.9701 , then diagnosis = M

If area2 < 21.2133 , then diagnosis = B
If 21.2133 <= area2 < 40.3371 , then diagnosis = B
If 72.7841 <= area2 , then diagnosis = M
If 40.3371 <= area2 < 72.7841 , then diagnosis = M

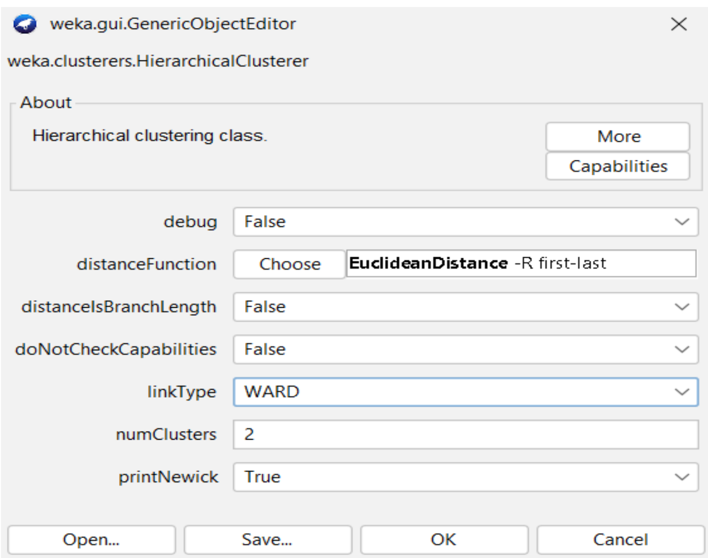
If symmetry1 < 0.1742 , then diagnosis = B
If 0.1742 <= symmetry1 < 0.1812 , then diagnosis = B
If 0.1931 <= symmetry1 , then diagnosis = M
If 0.1742 <= symmetry1 < 0.1931 , then diagnosis = M

If perimeter2 < 2.0063 , then diagnosis = B
If 2.0063 <= perimeter2 < 2.8661 , then diagnosis = B
If 4.3248 <= perimeter2 , then diagnosis = M
If 2.8661 <= perimeter2 < 4.3248, then diagnosis = M

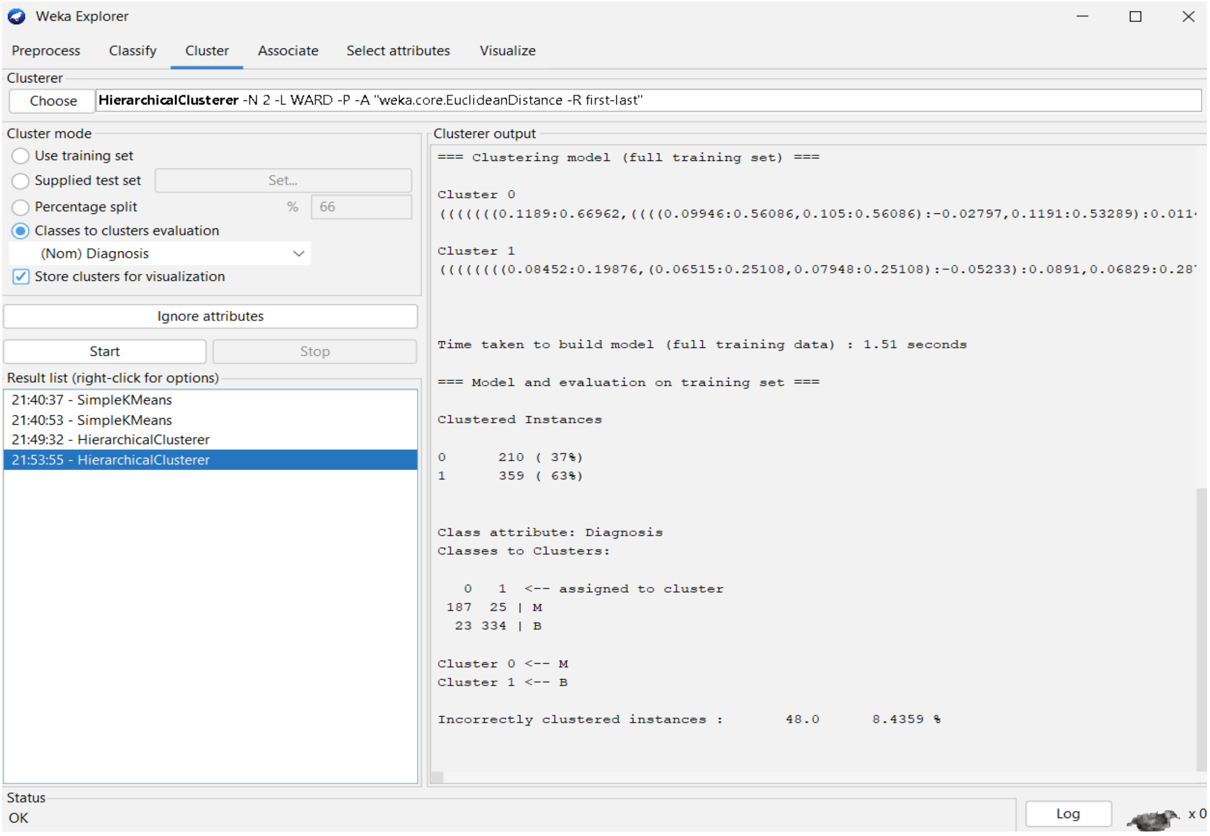
<u>Hierarchical Clustering</u>

Cluster mode: Classes to cluster evaluation

With Default parameter settings, the hierarchical clustering algorithm is not working well. When changing the linktype parameter to WARD the hierarchical clustering method clusters the instances with higher accuracy

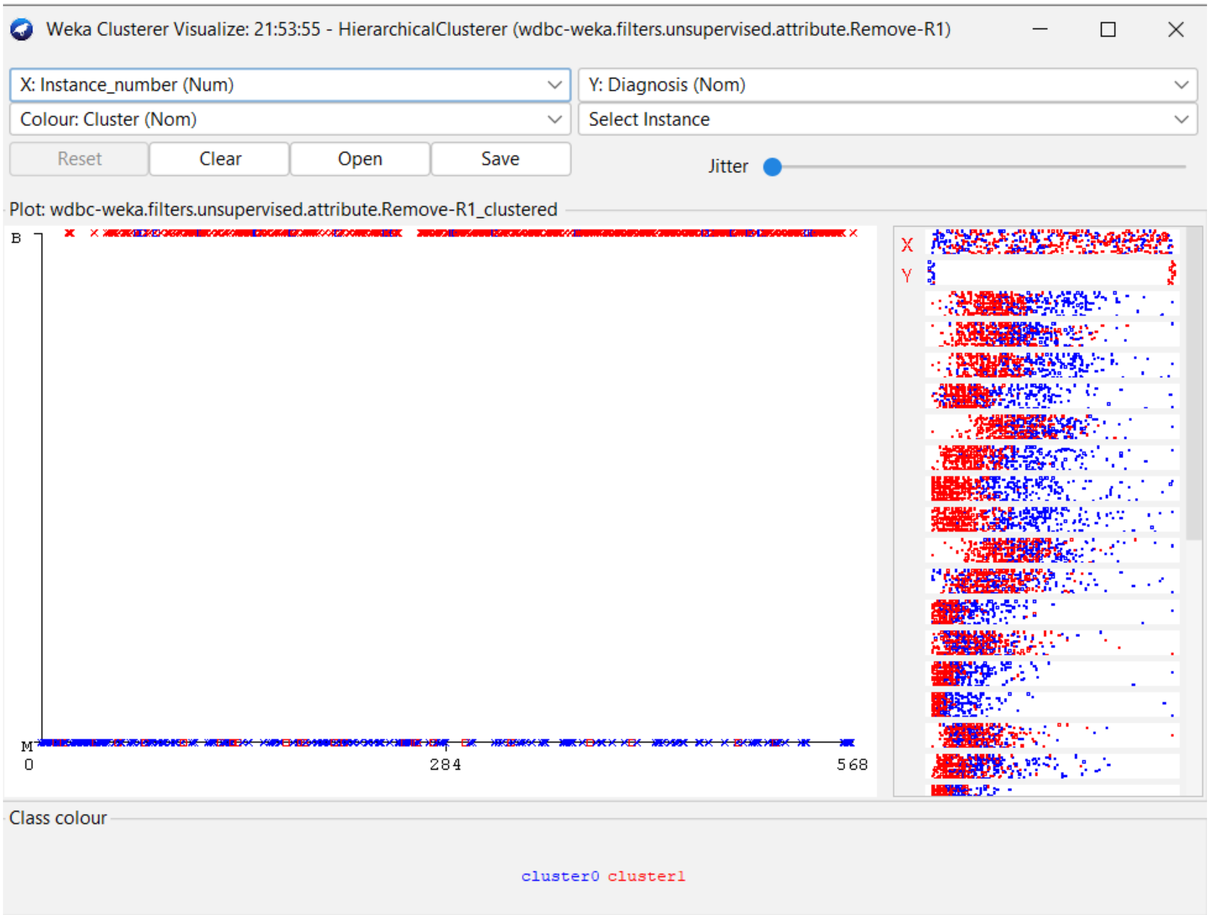


<u>Output</u>



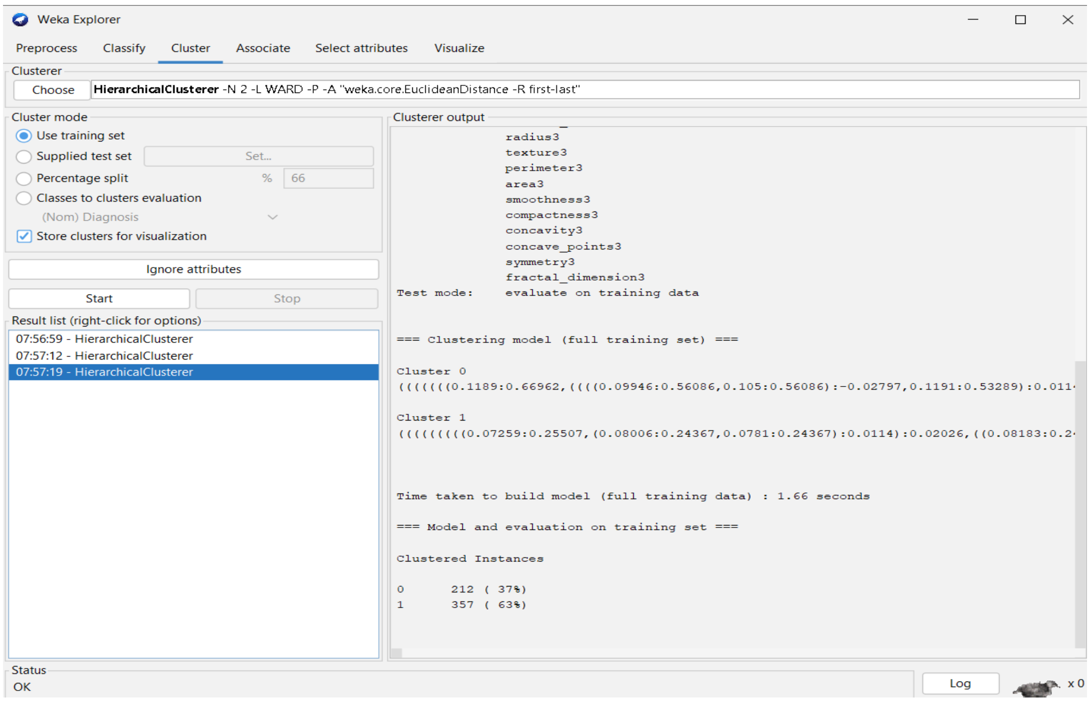
Accuracy (Correctly clustered instances) : 100 - 8.4359 = 91.5641%

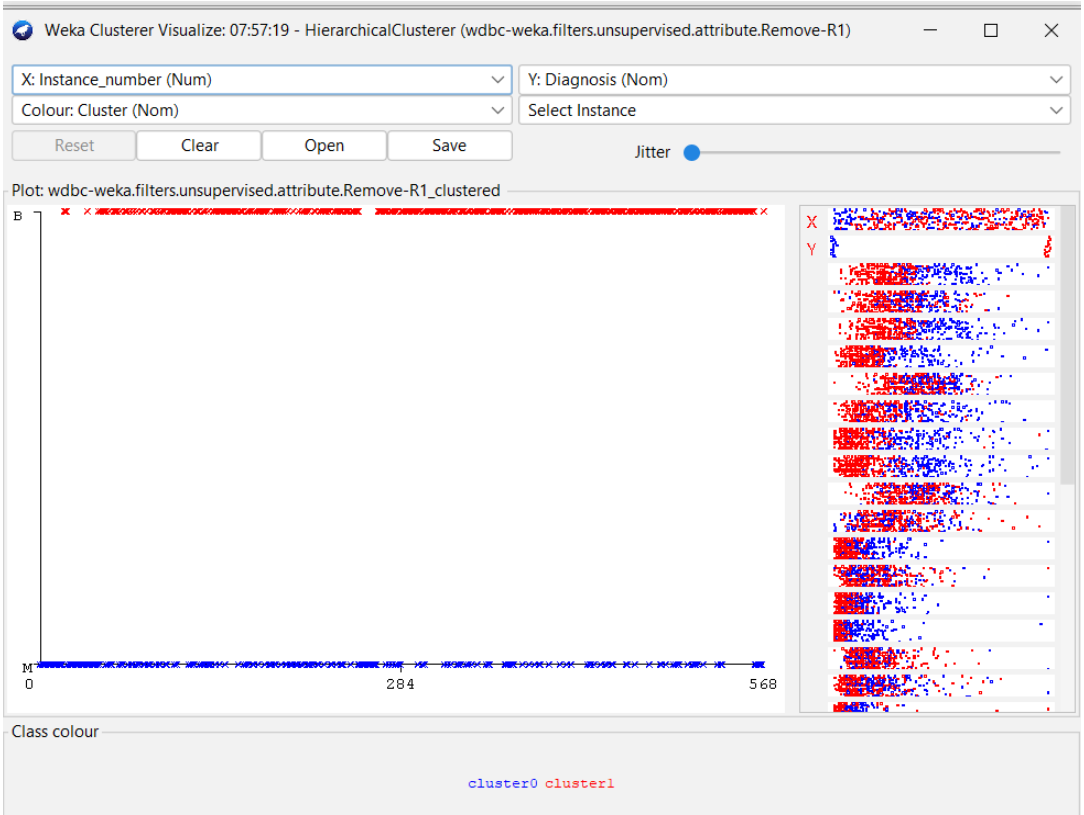
Time taken : 1.51 Seconds



Cluster mode: Use training set

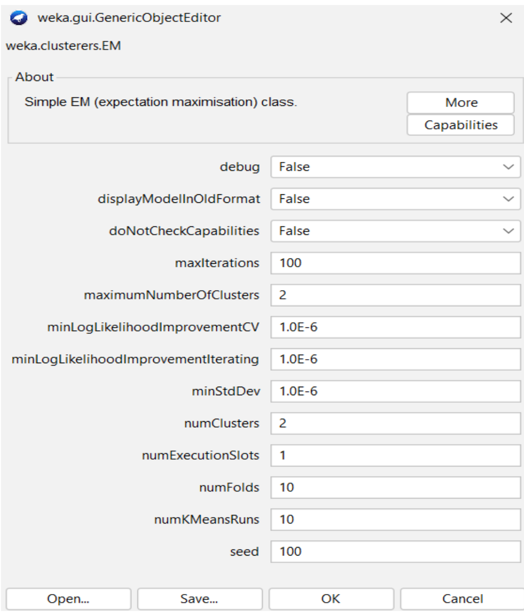
<u>Output</u>





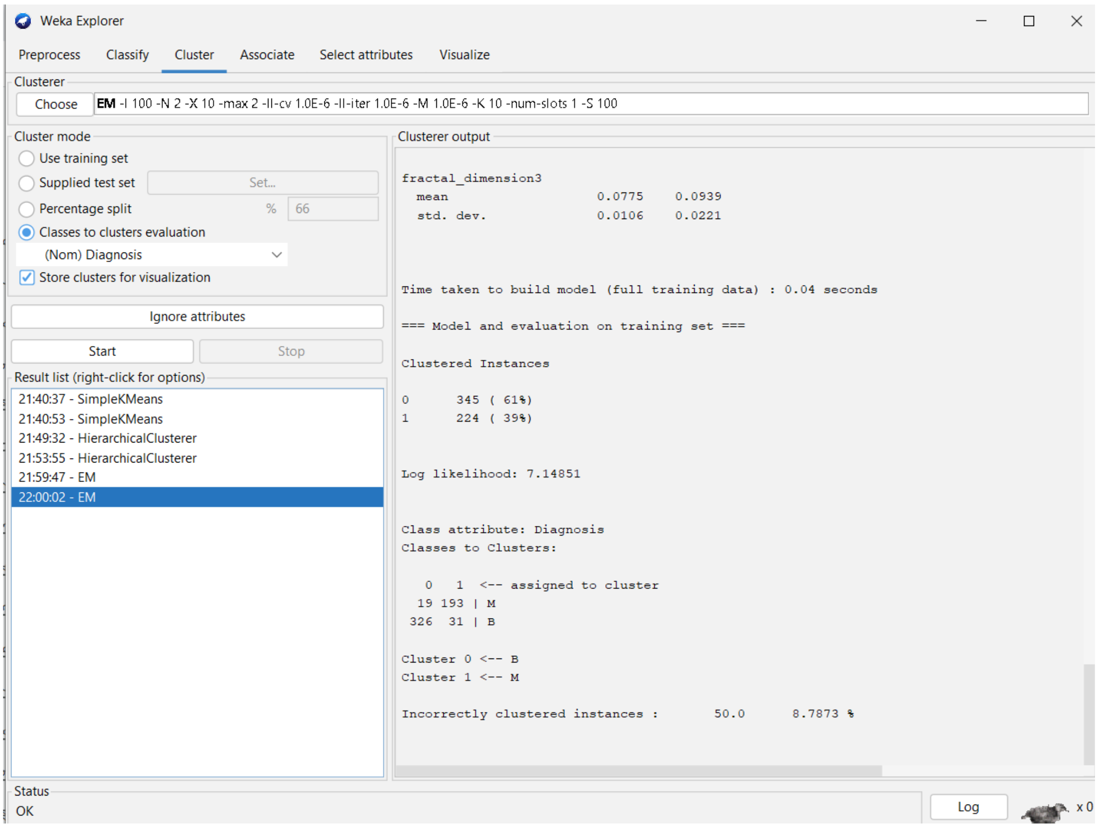
EM Clustering

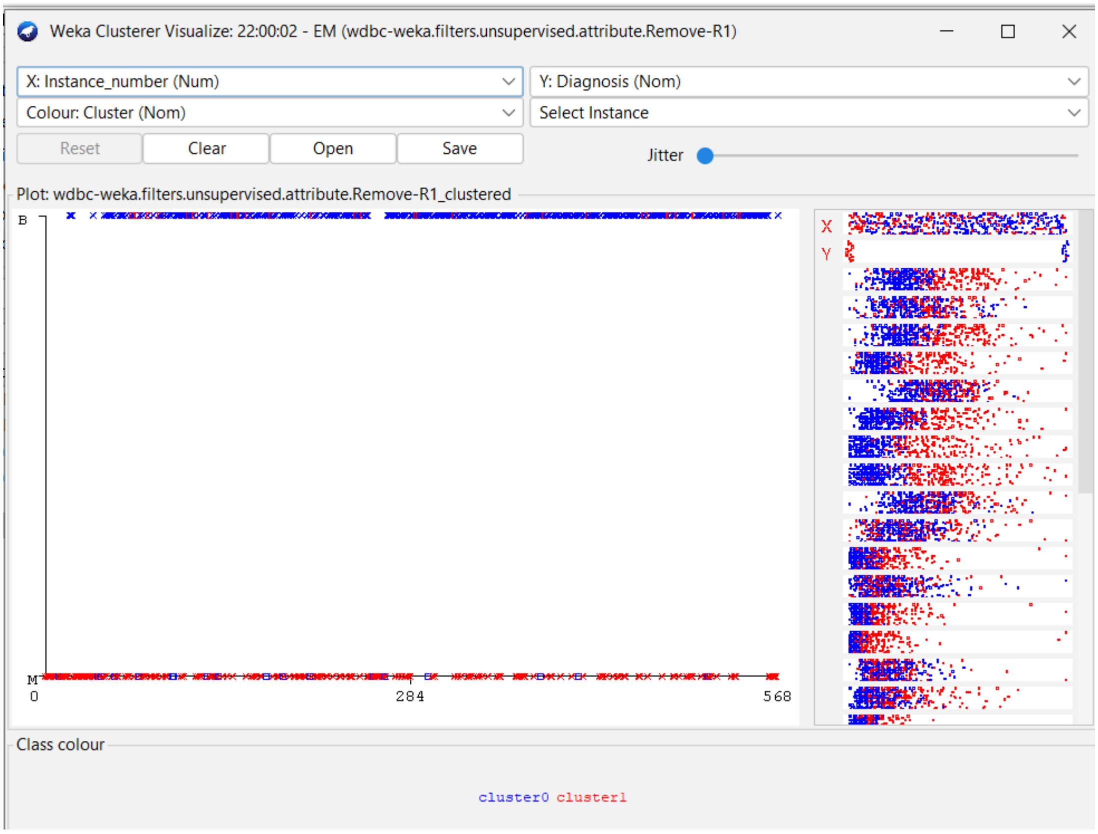
Default parameter setting



Output

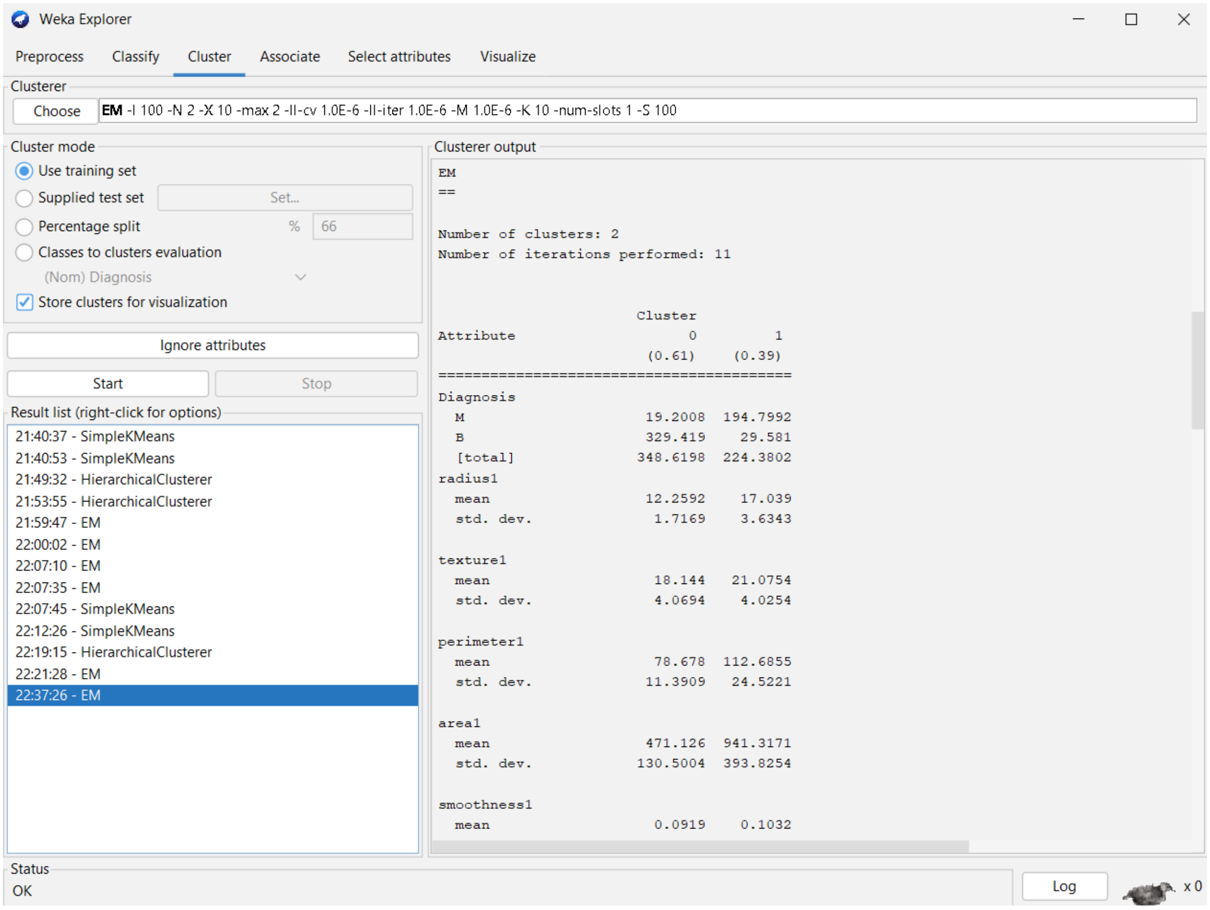
Cluster mode: Classes to cluster evaluation





Accuracy (Correctly clustered instances) : 100 - 8.7873 = 91.2127%
Time taken : 0.04 Sec

Cluster mode: Use training set




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

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

Clustered Instances

0       347 ( 61%)
1       222 ( 39%)


Log likelihood: 6.86608
```

Number of iteration : 11
Time taken : 0.02 seconds

Analysis of output

```
Clusterer output

                          Cluster
Attribute                       0          1
                           (0.61)     (0.39)
============================================
Diagnosis
  M                       19.2008   194.7992
  B                       329.419     29.581
  [total]                348.6198   224.3802
radius1
  mean                    12.2592     17.039
  std. dev.                1.7169     3.6343

texture1
  mean                     18.144    21.0754
  std. dev.                4.0694     4.0254

perimeter1
  mean                     78.678   112.6855
  std. dev.               11.3909    24.5221

area1
  mean                    471.126   941.3171
  std. dev.              130.5004   393.8254

smoothness1
  mean                     0.0919     0.1032
  std. dev.                0.0127     0.0133

compactness1
  mean                     0.0747     0.1505
  std. dev.                0.0263     0.0504

concavity1
  mean                     0.0391     0.1663
  std. dev.                0.0259      0.073
```

```
Clusterer output
concave_points1
  mean                     0.0245      0.087
  std. dev.                0.0141      0.034

symmetry1
  mean                     0.1721     0.1952
  std. dev.                0.0228      0.028

fractal_dimension1
  mean                     0.0617     0.0644
  std. dev.                0.0051     0.0091

radius2
  mean                     0.2778     0.6037
  std. dev.                0.1022     0.3398

texture2
  mean                     1.1991     1.2445
  std. dev.                0.5822     0.4978

perimeter2
  mean                     1.9333       4.32
  std. dev.                0.6933     2.4944

area2
  mean                     20.981    70.5071
  std. dev.                 8.123    60.7343

smoothness2
  mean                     0.0069     0.0073
  std. dev.                0.0028     0.0033

compactness2
  mean                     0.0179     0.0373
  std. dev.                0.0099     0.0209
```

```
Clusterer output
concavity2
  mean                     0.0203       0.05
  std. dev.                0.0142     0.0384

concave_points2
  mean                     0.0089     0.0163
  std. dev.                 0.004     0.0063

symmetry2
  mean                     0.0199     0.0215
  std. dev.                0.0067     0.0101

fractal_dimension2
  mean                     0.0031      0.005
  std. dev.                0.0015     0.0035

radius3
  mean                    13.5631    20.4871
  std. dev.                1.9954      4.925

texture3
  mean                    23.9885    28.3094
  std. dev.                5.6806     5.9006

perimeter3
  mean                    87.9838   137.3086
  std. dev.               13.4648    33.4565

area3
  mean                   575.1572  1356.6448
  std. dev.              168.8173   641.5027

smoothness3
  mean                     0.1249      0.144
  std. dev.                0.0194      0.023
```

```
Clusterer output
  mean                     0.1249      0.144
  std. dev.                0.0194      0.023

compactness3
  mean                     0.1723     0.3821
  std. dev.                0.0778     0.1642

concavity3
  mean                     0.1511     0.4609
  std. dev.                0.1017     0.1914

concave_points3
  mean                     0.0735     0.1787
  std. dev.                0.0349     0.0489

symmetry3
  mean                     0.2704     0.3208
  std. dev.                0.0406     0.0753

fractal_dimension3
  mean                     0.0775     0.0939
  std. dev.                0.0107     0.0221
```

If we closely see the mean values of each attribute it is almost the same value as the centroid values calculated in the KMeans cluster. Based on these values also we can create conditions and use it for decision making in future.

Some conditions derived from cluster output

If radius1 is 12.2592 or near to that value then there is high possibility that diagnosis is B
If radius1 is 17.039 or near to that value then there is high possibility that diagnosis is M

If area1 is 471.126 or near to that value then there is high possibility that diagnosis is B
If area1 is 941.3171 or near to that value then there is high possibility that diagnosis is M

Comparison

Cluster mode : Classes to cluster evaluation

	KMean	Hierarchical	EM
No of clusters	2	2	2
Accuracy	92.7944 %	91.5641%	91.2127%
Time taken	0 Sec	1.51 Sec	0.04 Sec

Cluster mode : Using training set

	KMean	Hierarchical	EM
No of clusters	2	2	2
Iteration	4	-	11
Sum of squared Errors	228.7944%	-	-
Time taken	0 Sec	1.66 Sec	0.02 Sec

Conclusion

In conclusion, for the breast cancer diagnostic dataset simple KMeans clustering algorithms performs better than other 2 clustering algorithms with correctly identified instances percentage 92.7944 %. Also KMeans clustering algorithm takes less time than other 2 with near to 0 seconds. Next both hierarchical clustering and EM cluster works with almost the same accuracy (Hierarchical : 91.5641% , EM: 91.2127%). But the time taken for hierarchical clustering is higher than EM clustering. (Hierarchical : 1.51 Sec , EM: 0.04 Sec).

When selecting cluster mode using a training set, we can find out hidden and useful information by analysing cluster outputs. The centroid values we got from KMeans clustering and Mean values got from EM clustering are almost the same. We can create some conditions from those values and can use it for decision making for new instances.

Task 02

Dataset : Online Retail Dataset

Dataset description : This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

Attributes

InvoiceNo:. Nominal
StockCode: Nominal
Description: Nominal
Quantity: Numeric
InvoiceDate: Numeric
UnitPrice: Numeric
CustomerID: Nominal
Country: Nominal

Objectives

By using this dataset, we can find out the interesting associations between the stock items. By finding that we can do various activities that make the company more profitable. First one is product bundling, which means the frequently purchased items can be identified and can be bundled together with a discount to increase sales. Next one is cross-selling opportunities. For example if we find associations between items A,B that are purchased together in many transactions, we can recommend B to a user, if the user tries to buy A and the chance of the user to buy B is very high. Another important benefit is selling less frequently selling products along with frequent selling items. For example if we find associations between items A,B that are purchased together in many transactions, we can make a promotion like if the user buys A and B together the C can be purchased with a 20% discount. Here item C is less frequently selling items. Since users are interested to buy A and B together, there is a high chance to buy C too with a discount.

Preparation and Preprocessing

Before association rule mining there are preprocessing steps that should be done.

Load dataset in jupyter notebook

In [1]:
```
import pandas as pd

#load dataset
df = pd.read_csv("C:\\Users\\USER\\Desktop\\Online Retail\\online_retail.csv")
df.head()
```

Out[1]:

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	12/1/2010 8:26	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	12/1/2010 8:26	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	12/1/2010 8:26	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	12/1/2010 8:26	3.39	17850.0	United Kingdom

Remove the attributes and its values which are not needed for association rule mining

In [2]:
```
#drop unneccessary columns
df2 = df.drop(columns=['Description', 'Quantity','InvoiceDate', 'CustomerID','UnitPrice', 'Country'])
```

Reasons for removing above attributes

Description: Description is not necessary for association rule mining and it will not give any useful associations

Quantity: This attribute has numeric values, the Weka apriori algorithm works only with nominal data.

InvoiceDate: Invoice date will not help to find useful rules.

CustomerID: For Association rule mining purpose we are interested in every transaction or invoice, not in individual users. So customerID is not needed.

UnitPrice: This attribute has numeric values, the Weka apriori algorithm works only with nominal data.

Country: We consider all transactions in the dataset, not only for specific countries' customers. So Country can be ignored.

Finding the total number of purchases of each stock individually.

```
In [3]: #print how many times perticular stock purchased
        stockcode_counts  = df2['StockCode'].value_counts()
        stockcode_counts

Out[3]: 85123A    2313
        22423     2203
        85099B    2159
        47566     1727
        20725     1639
                  ...
        84596g       1
        90091        1
        21653        1
        21431        1
        20849        1
        Name: StockCode, Length: 4070, dtype: int64
```

Storing those values in a variable called 'stockcode_value_counts'
Defining a threshold value
FIlter the rows that have stock items totally purchased less than threshold value. By doing this we can remove less frequent items. If we keep the less frequent items it can give unwanted rules in weka.

```
In [5]: #store it in a variable
        stockcode_value_counts = df2['StockCode'].value_counts()

In [6]: #define a threshold, By that we can remove very less frequent items
        threshold = 1600

In [8]: #filtering rows with stocks that purchased more than threshold value in overall purchase count
        filtered_rows = df2[df2['StockCode'].map(stockcode_value_counts) > threshold]
```

Convert the dataframe in the suitable form to do association rule mining in weka

```
In [9]: #convert the dataframe that suitable for assosiate rule mining
pivot_df = filtered_rows.pivot_table(index='InvoiceNo', columns='StockCode', aggfunc='size', fill_value=0)

# Convert values to True/False
pivot_df = pivot_df.applymap(lambda x: x > 0)

# Reset index to make 'InvoiceNo' a column
pivot_df.reset_index(inplace=True)

print(pivot_df)
```

```
StockCode InvoiceNo  20725  22423  47566  85099B  85123A
0            536365  False  False  False   False    True
1            536373  False  False  False   False    True
2            536375  False  False  False   False    True
3            536378   True  False  False   False   False
4            536386  False  False  False    True   False
...             ...    ...    ...    ...     ...     ...
7124        C581117  False  False  False    True   False
7125        C581128  False  False  False    True   False
7126        C581228  False   True  False   False   False
7127        C581229  False  False  False    True   False
7128        C581235  False   True  False   False   False

[7129 rows x 6 columns]
```

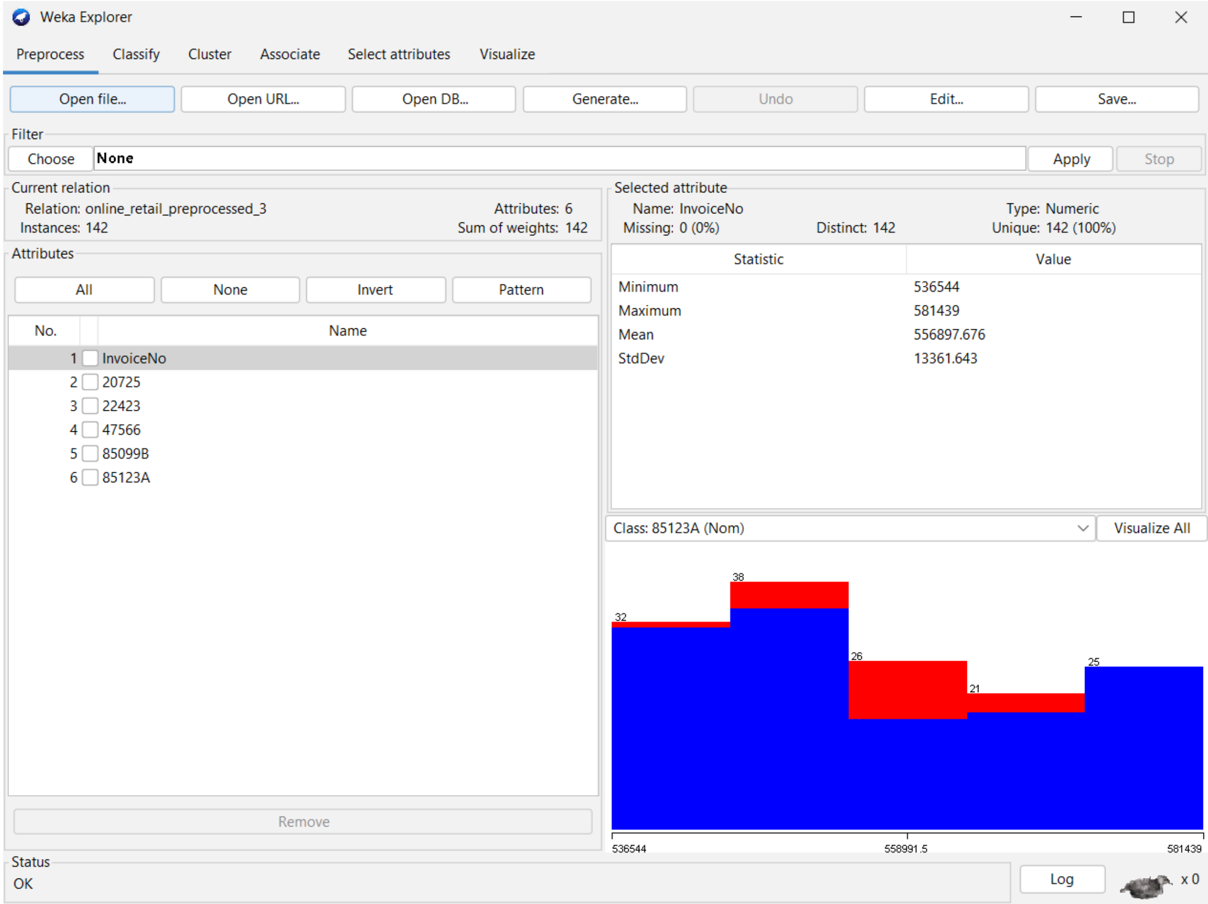
Next, we need to get the transaction with multiple stock purchases. By doing this we can remove transactions with 1 or 2 stock items, which can lead weka to give unwanted rules.

```
In [10]: #Getting the number of items purchased in each transaction
true_counts = pivot_df.iloc[:, 1:].sum(axis=1)

In [11]: #Filtering transactions in which, user purchased more than 3 stock item
filtered_rows = pivot_df[true_counts > 3]

In [12]: filtered_rows.shape

Out[12]: (142, 6)
```

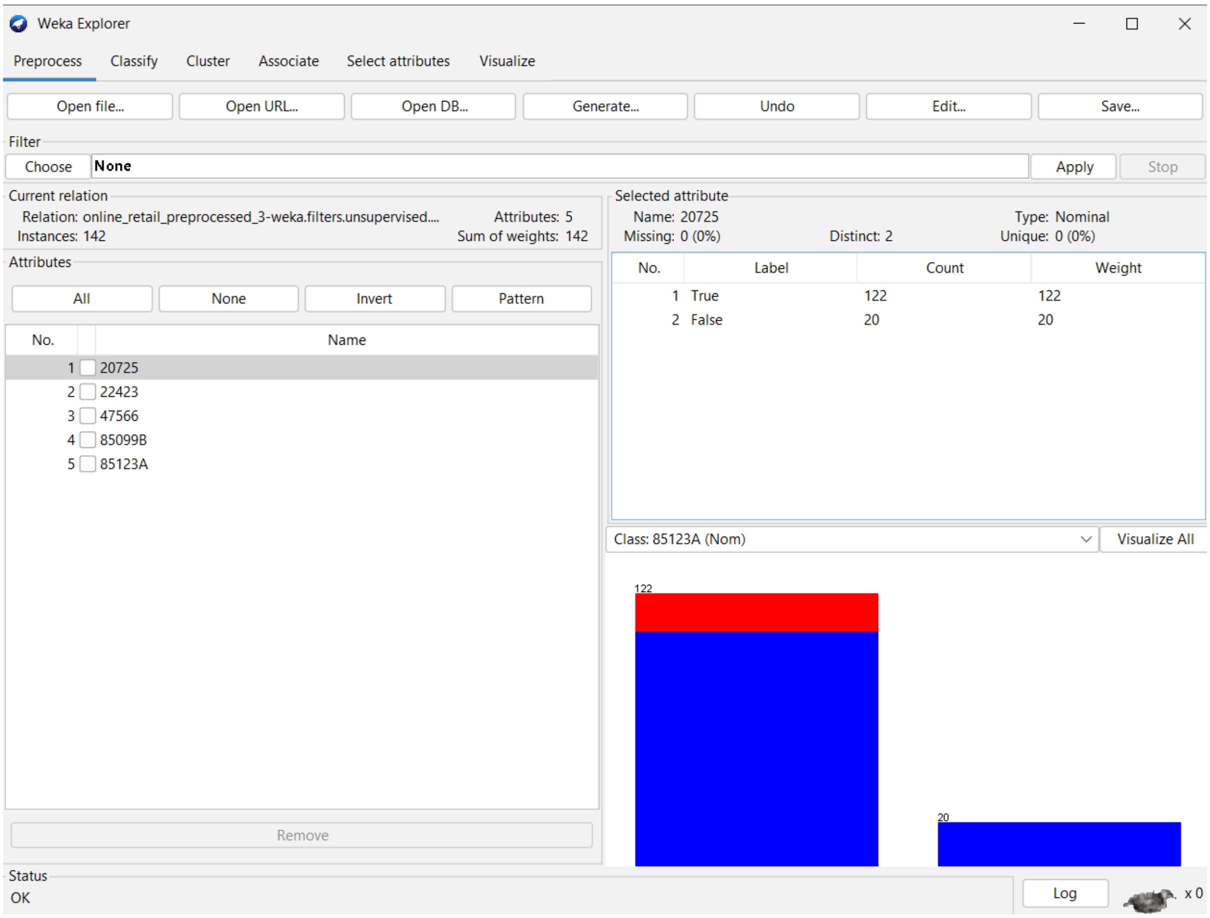
Now, the preprocessing is done and can save the file.

```
In [13]: #Save the processed file
file_path = 'C:\\Users\\USER\\Desktop\\Online Retail\\online_retail_preprocessed_3.csv'
filtered_rows.to_csv(file_path, index=False)
```

Now load the preprocessed dataset into weka

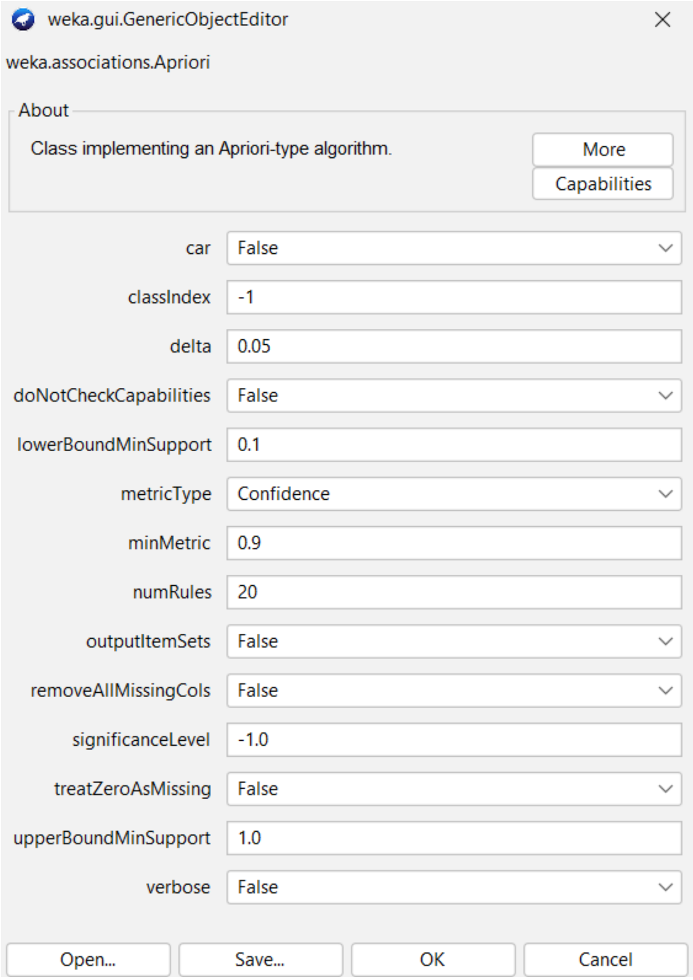


InvoiceNo attribute is unnecessary for association rule mining. So it can be removed

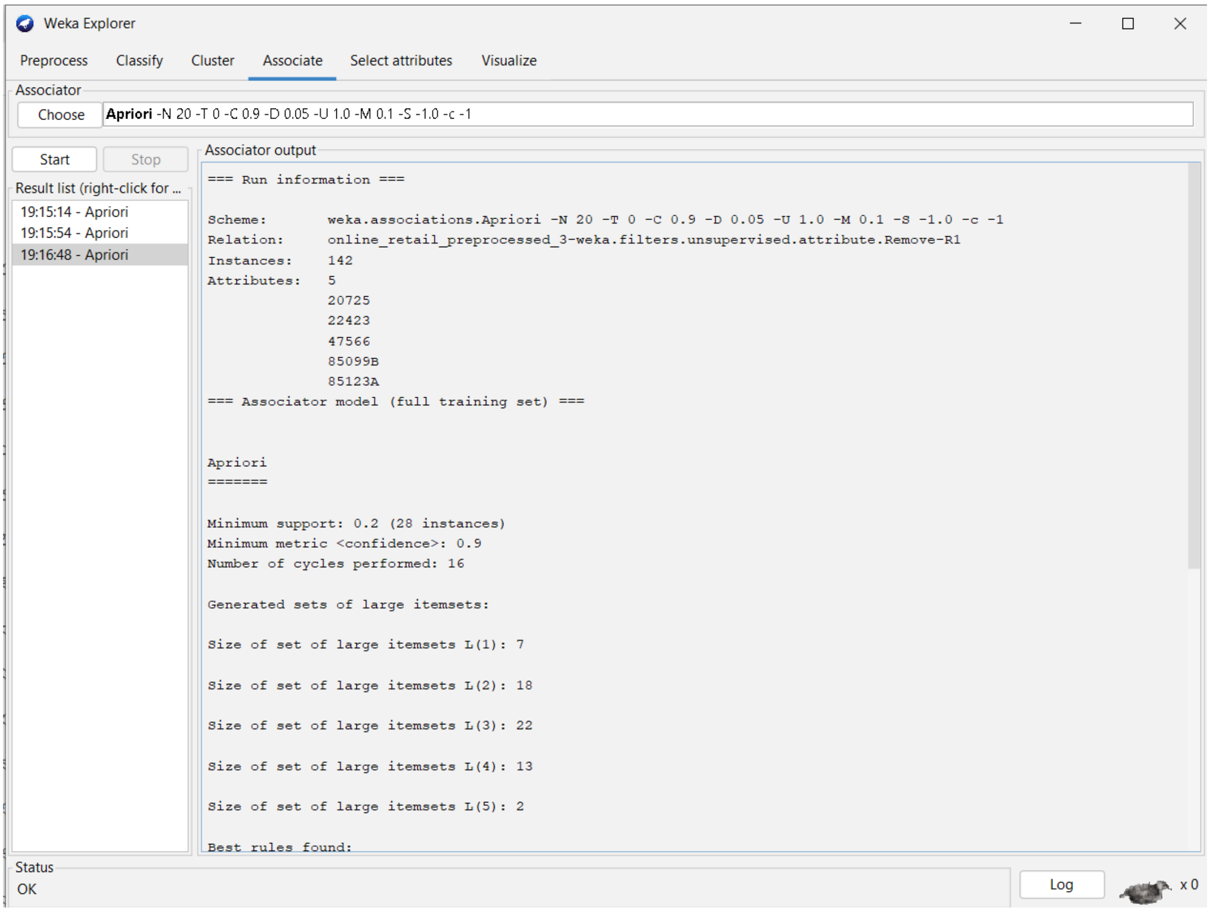

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: online_retail_preprocessed_3-weka.filters.unsupervised....
Attributes: 5
Instances: 142
Sum of weights: 142
Attributes
All
None
Invert
Pattern
No.
Name
1 20725
2 22423
3 47566
4 85099B
5 85123A
Remove
Selected attribute
Name: 20725
Type: Nominal
Missing: 0 (0%)
Distinct: 2
Unique: 0 (0%)
No.
Label
Count
Weight
1 True 122 122
2 False 20 20
Class: 85123A (Nom)
Visualize All
122
20
Status
OK
Log
x 0

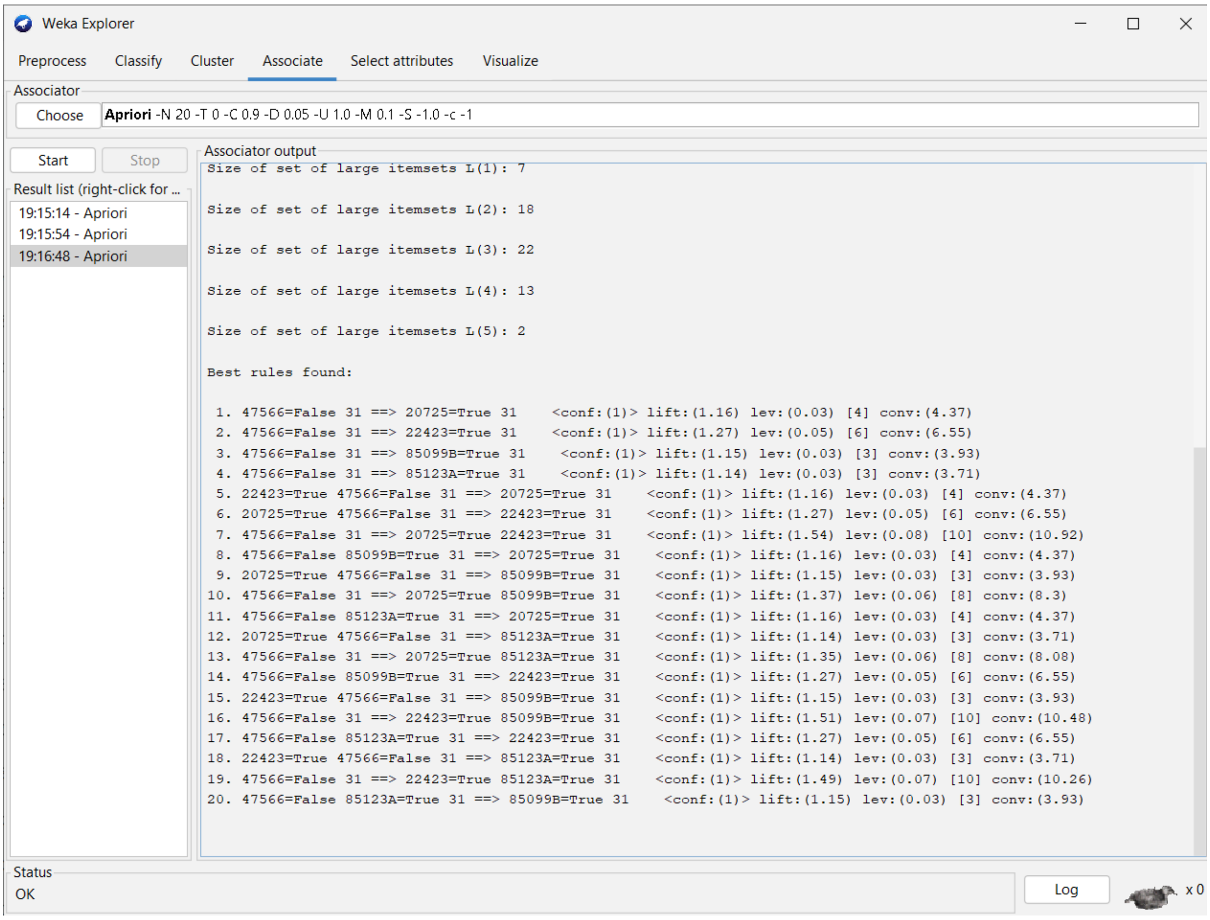
<u>Rule mining process</u>

Next, Select Associate tab and select apriori algorithm with following parameters



The minimum confidence is set to 0.9 , The output of weka is following


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 20 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-click for ...
19:15:14 - Apriori
19:15:54 - Apriori
19:16:48 - Apriori
Associator output
=== Run information ===
Scheme: weka.associations.Apriori -N 20 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: online_retail_preprocessed_3-weka.filters.unsupervised.attribute.Remove-R1
Instances: 142
Attributes: 5
20725
22423
47566
85099B
85123A
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.2 (28 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 16
Generated sets of large itemsets:
Size of set of large itemsets L(1): 7
Size of set of large itemsets L(2): 18
Size of set of large itemsets L(3): 22
Size of set of large itemsets L(4): 13
Size of set of large itemsets L(5): 2
Best rules found:
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 20 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-click for ...
19:15:14 - Apriori
19:15:54 - Apriori
19:16:48 - Apriori
Associator output
Size of set of large itemsets L(1): 7
Size of set of large itemsets L(2): 18
Size of set of large itemsets L(3): 22
Size of set of large itemsets L(4): 13
Size of set of large itemsets L(5): 2
Best rules found:
1. 47566=False 31 ==> 20725=True 31 <conf:(1)> lift:(1.16) lev:(0.03) [4] conv:(4.37)
2. 47566=False 31 ==> 22423=True 31 <conf:(1)> lift:(1.27) lev:(0.05) [6] conv:(6.55)
3. 47566=False 31 ==> 85099B=True 31 <conf:(1)> lift:(1.15) lev:(0.03) [3] conv:(3.93)
4. 47566=False 31 ==> 85123A=True 31 <conf:(1)> lift:(1.14) lev:(0.03) [3] conv:(3.71)
5. 22423=True 47566=False 31 ==> 20725=True 31 <conf:(1)> lift:(1.16) lev:(0.03) [4] conv:(4.37)
6. 20725=True 47566=False 31 ==> 22423=True 31 <conf:(1)> lift:(1.27) lev:(0.05) [6] conv:(6.55)
7. 47566=False 31 ==> 20725=True 22423=True 31 <conf:(1)> lift:(1.54) lev:(0.08) [10] conv:(10.92)
8. 47566=False 85099B=True 31 ==> 20725=True 31 <conf:(1)> lift:(1.16) lev:(0.03) [4] conv:(4.37)
9. 20725=True 47566=False 31 ==> 85099B=True 31 <conf:(1)> lift:(1.15) lev:(0.03) [3] conv:(3.93)
10. 47566=False 31 ==> 20725=True 85099B=True 31 <conf:(1)> lift:(1.37) lev:(0.06) [8] conv:(8.3)
11. 47566=False 85123A=True 31 ==> 20725=True 31 <conf:(1)> lift:(1.16) lev:(0.03) [4] conv:(4.37)
12. 20725=True 47566=False 31 ==> 85123A=True 31 <conf:(1)> lift:(1.14) lev:(0.03) [3] conv:(3.71)
13. 47566=False 31 ==> 20725=True 85123A=True 31 <conf:(1)> lift:(1.35) lev:(0.06) [8] conv:(8.08)
14. 47566=False 85099B=True 31 ==> 22423=True 31 <conf:(1)> lift:(1.27) lev:(0.05) [6] conv:(6.55)
15. 22423=True 47566=False 31 ==> 85099B=True 31 <conf:(1)> lift:(1.15) lev:(0.03) [3] conv:(3.93)
16. 47566=False 31 ==> 22423=True 85099B=True 31 <conf:(1)> lift:(1.51) lev:(0.07) [10] conv:(10.48)
17. 47566=False 85123A=True 31 ==> 22423=True 31 <conf:(1)> lift:(1.27) lev:(0.05) [6] conv:(6.55)
18. 22423=True 47566=False 31 ==> 85123A=True 31 <conf:(1)> lift:(1.14) lev:(0.03) [3] conv:(3.71)
19. 47566=False 31 ==> 22423=True 85123A=True 31 <conf:(1)> lift:(1.49) lev:(0.07) [10] conv:(10.26)
20. 47566=False 85123A=True 31 ==> 85099B=True 31 <conf:(1)> lift:(1.15) lev:(0.03) [3] conv:(3.93)
Status
OK
Log
x 0

Resulting Rules

Here we can get some useful associations.

1. If the user is purchasing stock item 22423 and not purchasing stock item 47566, then he/she will purchase the stock item 20725 with confidence 1.
2. If the user is purchasing stock item 85099B and not purchasing stock item 47566, then he/she will purchase the stock item 20725 with confidence 1.
3. If the user is purchasing stock item 85123A and not purchasing stock item 47566, then he/she will purchase the stock item 20725 with confidence 1.
4. If the user is purchasing stock item 85099B and not purchasing stock item 47566, then he/she will purchase the stock item 22423 with confidence 1.
5. If the user is purchasing stock item 85123A and not purchasing stock item 47566, then he/she will purchase the stock item 22423 with confidence 1.
6. If the user is purchasing stock item 85123A and not purchasing stock item 47566, then he/she will purchase the stock item 85099B with confidence 1.

Recommendations

1. When a customer purchases stock item 22423, recommend stock item 20275 on the website.
2. When a customer purchases stock item 85099B, recommend stock item 20275 on the website.
3. When a customer purchases stock item 85123A, recommend stock item 20275 on the website.
4. When a customer purchases stock item 85099B, recommend stock item 22423 on the website.
5. Make a product bundle with stock item 22423, and stock item 20275 and give a discount for product bundle purchase, which means the customer will get a discount when he/she purchases both stocks together.
6. Make a product bundle with stock item 85123A , and stock item 85099B and give a discount for product bundle purchase, which means the customer will get a discount when he/she purchases both stocks together.
7. Select a less frequent sell item X relates to stock item 85123A and stock item 22423 and start promotion as if a customer purchases both stock item 85123A and 22423, then can purchase X with a particular discount.