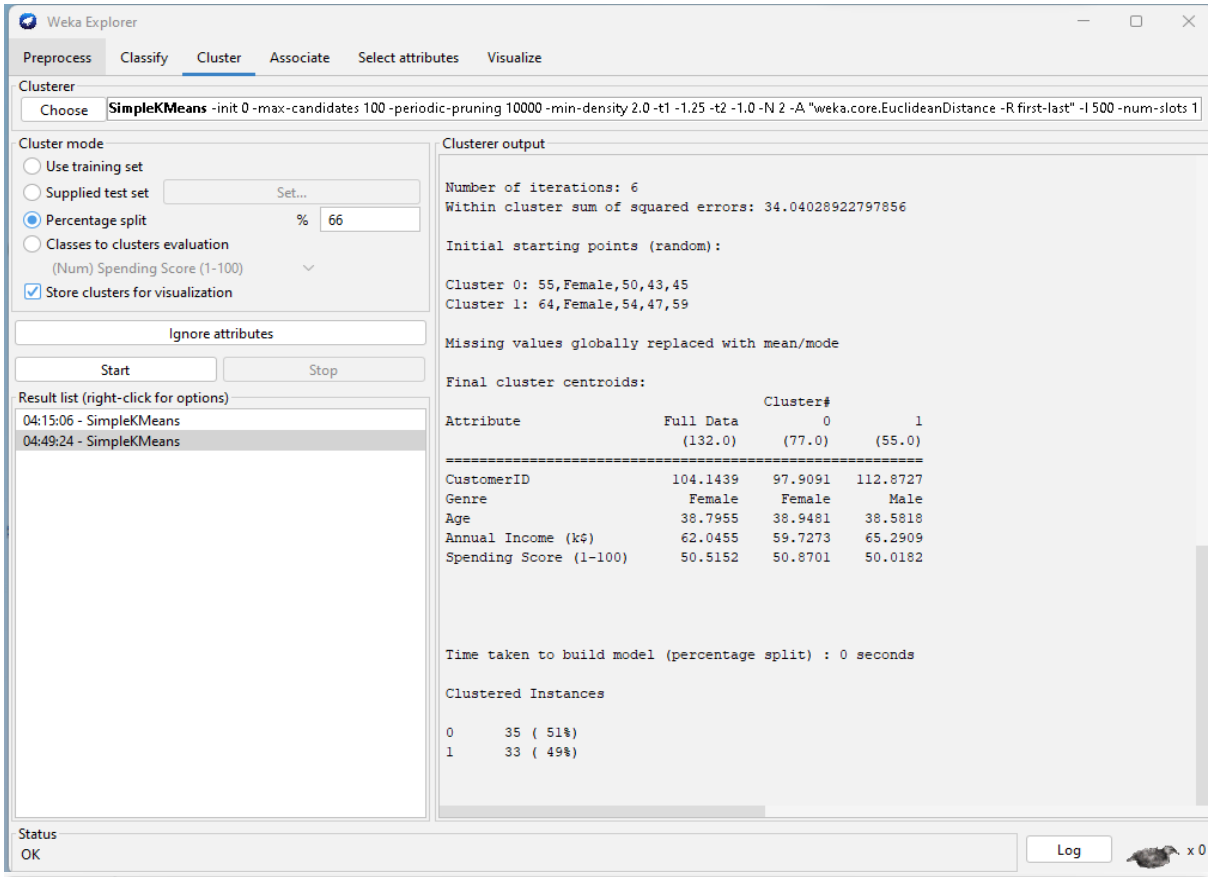
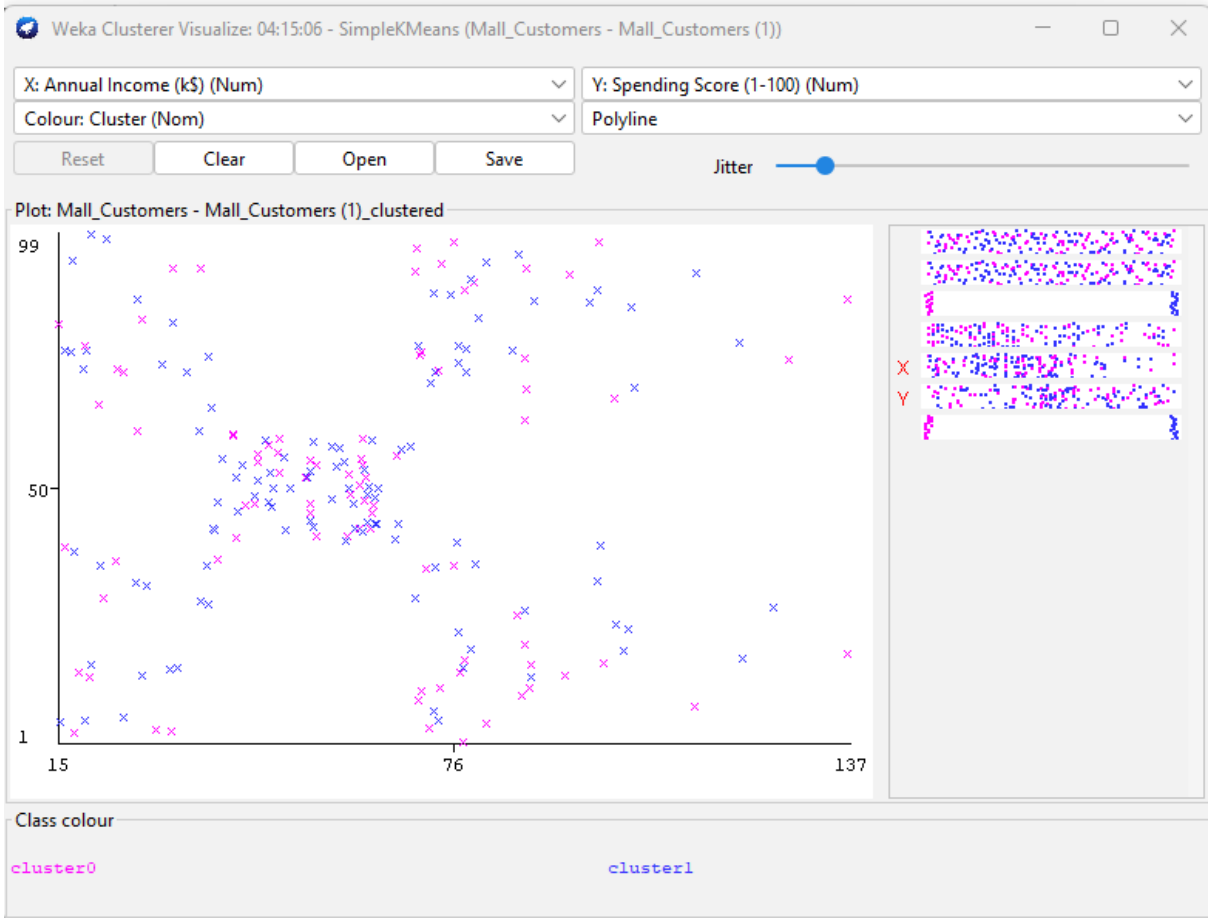

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Clusterer
Choose
SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1
Cluster mode
Use training set
Supplied test set
Set...
Percentage split
%
66
Classes to clusters evaluation
(Num) Spending Score (1-100)
Store clusters for visualization
Ignore attributes
Start
Stop
Result list (right-click for options)
04:15:06 - SimpleKMeans
04:49:24 - SimpleKMeans
Clusterer output
Number of iterations: 6
Within cluster sum of squared errors: 34.04028922797856
Initial starting points (random):
Cluster 0: 55,Female,50,43,45
Cluster 1: 64,Female,54,47,59
Missing values globally replaced with mean/mode
Final cluster centroids:
Cluster#
Attribute Full Data 0 1
(132.0) (77.0) (55.0)
CustomerID 104.1439 97.9091 112.8727
Genre Female Female Male
Age 38.7955 38.9481 38.5818
Annual Income (k$) 62.0455 59.7273 65.2909
Spending Score (1-100) 50.5152 50.8701 50.0182
Time taken to build model (percentage split) : 0 seconds
Clustered Instances
0 35 (51%)
1 33 (49%)
Status
OK
Log
x 0


Weka Clusterer Visualize: 04:15:06 - SimpleKMeans (Mall_Customers - Mall_Customers (1))
X: Annual Income (k$) (Num)
Y: Spending Score (1-100) (Num)
Colour: Cluster (Nom)
Polyline
Reset
Clear
Open
Save
Jitter
Plot: Mall_Customers - Mall_Customers (1)_clustered
99
50
1
15
76
137
X
Y
Class colour
cluster0
cluster1

2. Decision Tree Induction using WEKA

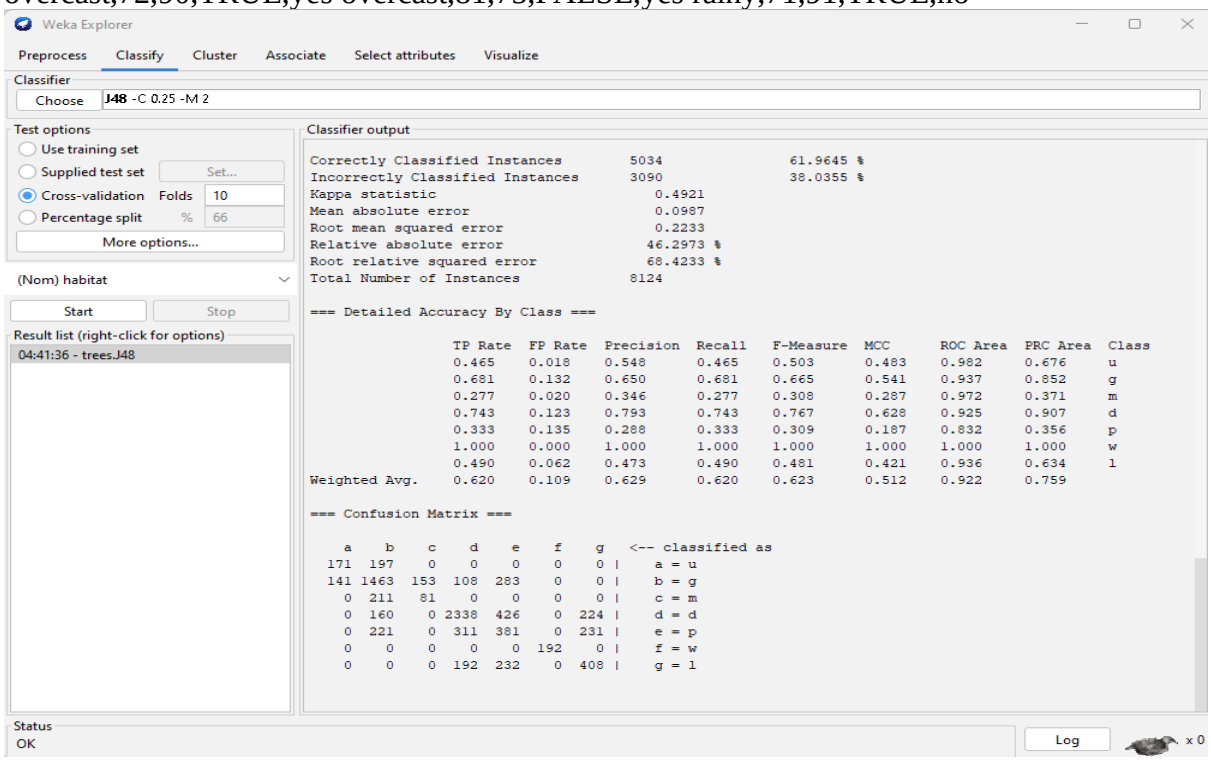
A decision tree is a flowchart like tree structure, where each internal node(non-leaf node) denotes a test on an attribute,each branch represents an outcome of the test,and each leaf node (or terminal node) holds a class label. The topmost node in a tree is the root node

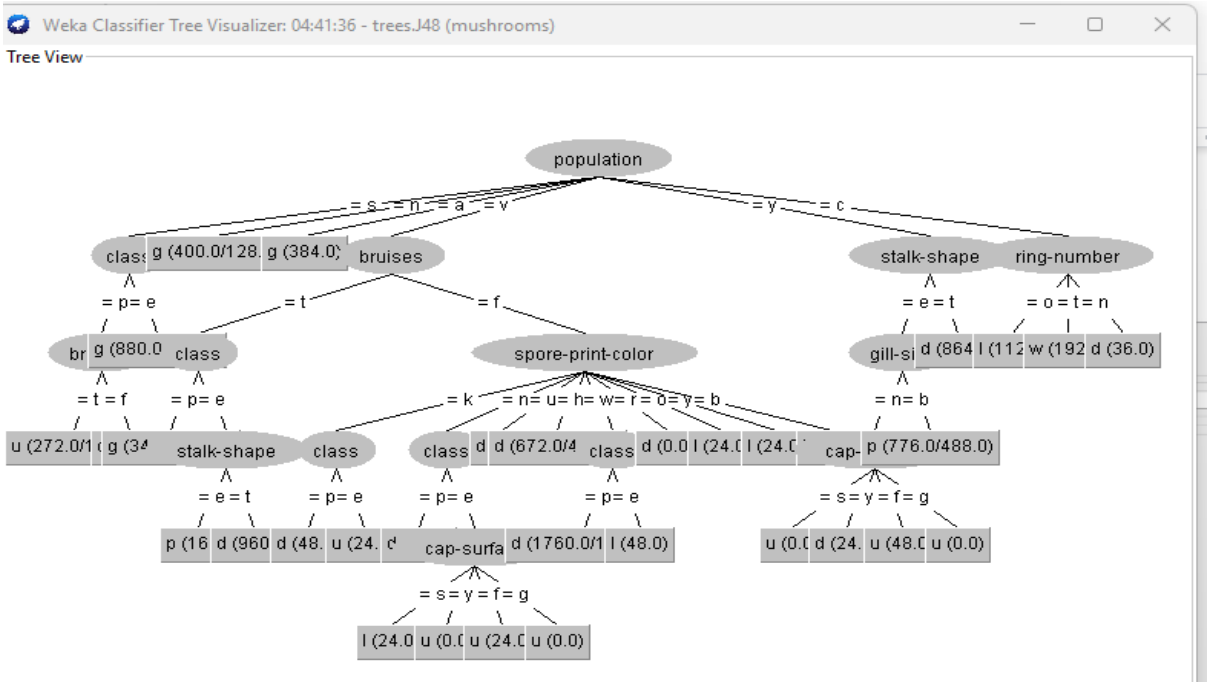
Example:-

Outlook	Temperature	Humidity	Windy	Class
sunny	hot	high	false	N
sunny	hot	high	true	N
overcast	hot	high	false	P
rain	mild	high	false	P
rain	cool	normal	false	P
rain	cool	normal	true	N
overcast	cool	normal	true	P
sunny	mild	high	false	N
sunny	cool	normal	false	P
rain	mild	normal	false	P
sunny	mild	normal	true	P
overcast	mild	high	true	P
overcast	hot	normal	false	P
rain	mild	high	true	N

Code:

@relation weather
@attribute outlook {sunny, overcast, rainy} @attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE} @attribute play {yes, no}
@data sunny,85,85,FALSE,no
sunny,80,90,TRUE,no overcast,83,86,FALSE,yes rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes rainy,65,70,TRUE,no overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no sunny,69,70,FALSE,yes rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes overcast,81,75,FALSE,yes rainy,71,91,TRUE,no





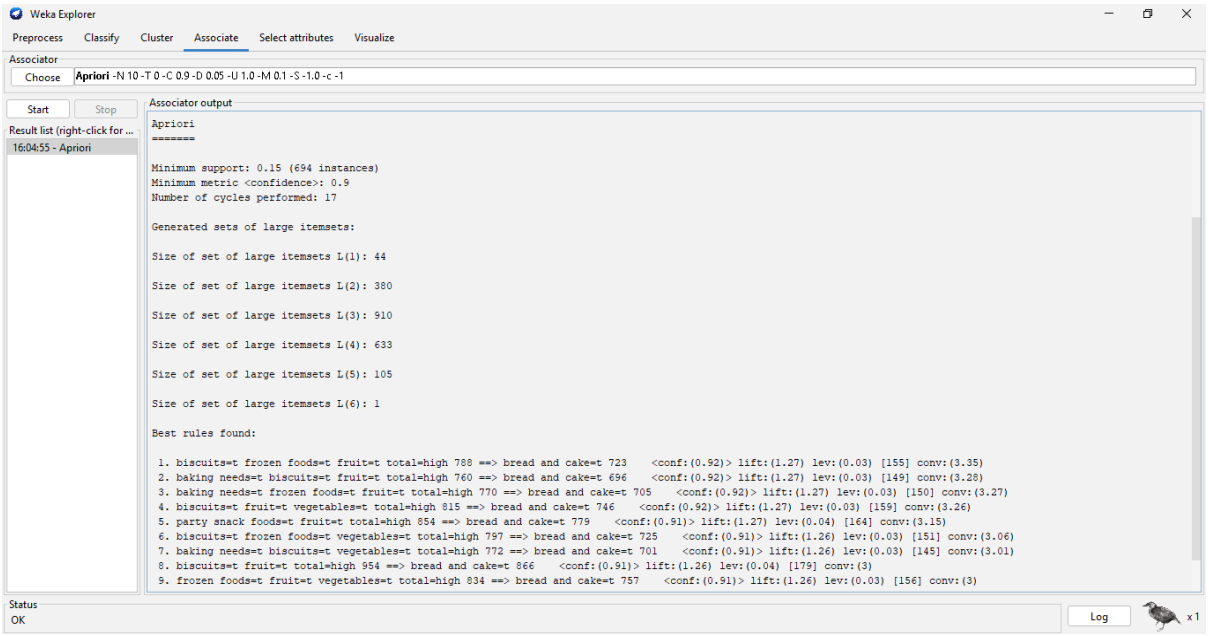
3. Apriori Algorithm using WEKA

In this current world, globalization is the main feature of any environment. Everyone has to be update, fast and forward and information is the main element for it. For survival in this world it's the basic need to use and to store the information means to prepare a proper database or dataset to analyze. Using and storing the database is not an issue, but finding the relevant dataset or to analyze the meaningful dataset for a particular aspect, from the junkyard of the database is very big problem in analysis of a specific part of the database. To solve this problem the concept of data mining is used to abstracts the desirable information. Useful information from the large databases has been extracted in the form of the association rules. There are many algorithms have been developed to extract the association rules from the large databases. Apriori algorithm is the most popular algorithm to extract the association rules from the databases.

TID	Items
1	A,B,C,D,G,H
2	A,B,C,D,E,F,H
3	B,C,D,E,H
4	B,E,G,H
5	A,B,D,E,G,H
6	A,C,F,G,H
7	B,D,E,G,H
8	A,C,D,E,G,H
9	B,C,D,E,H
10	A,C,E,F,H
11	C,E,H
12	A,D,E,F,H
13	B,C,E,F,H
14	A,B,C,F,H
15	A,B,E,F,H

Example

CODE:
@relation TEST_ITEM_TRANS @attribute A {TRUE, FALSE} @attribute B {TRUE, FALSE} @attribute C {TRUE, FALSE} @attribute D {TRUE, FALSE} @attribute E {TRUE, FALSE} @attribute F {TRUE, FALSE} @attribute G {TRUE, FALSE} @attribute H {TRUE, FALSE} @data
TRUE,TRUE,TRUE,TRUE,FALSE,FALSE,TRUE,TRUE
TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,FALSE,TRUE
FALSE,TRUE,TRUE,TRUE,FALSE,FALSE,FALSE,TRUE
FALSE,TRUE,FALSE,FALSE,TRUE,FALSE,TRUE,TRUE
TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE
TRUE,FALSE,TRUE,FALSE,FALSE,TRUE,TRUE,TRUE
FALSE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE,TRUE
TRUE,FALSE,TRUE,TRUE,TRUE,FALSE,TRUE,TRUE
FALSE,TRUE,TRUE,TRUE,TRUE,FALSE,FALSE,TRUE
TRUE,FALSE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE
FALSE,FALSE,TRUE,FALSE,TRUE,FALSE,FALSE,TRUE
TRUE,FALSE,FALSE,TRUE,TRUE,TRUE,FALSE,TRUE
FALSE,TRUE,TRUE,FALSE,TRUE,TRUE,FALSE,TRUE
TRUE,TRUE,TRUE,FALSE,FALSE,TRUE,FALSE,TRUE TRUE,TRUE,FALSE,FALSE,
TRUE,TRUE,FALSE,TRUE



Conclusion: