EXPERIMENT1

Aim: Demonstrate the following data preprocessing tasks.

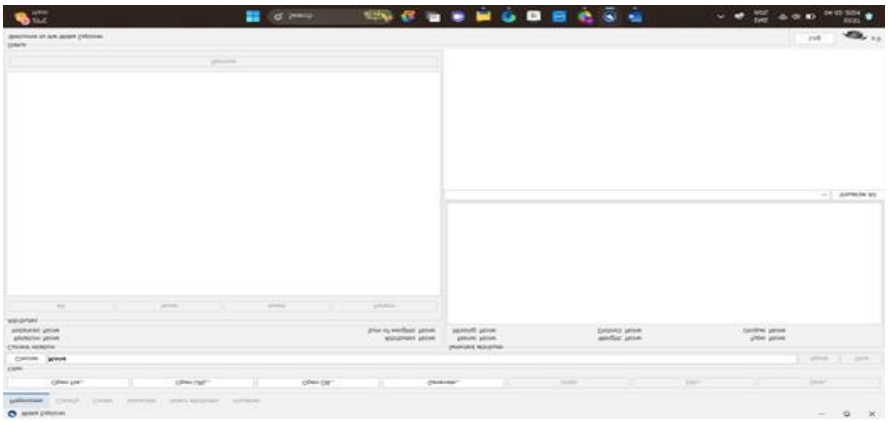
- ❖ Loading the dataset
- ❖ Adding Attribute
- ❖ Add expression
- ❖ Copy attribute
- ❖ Remove attribute

Description:WEKA - an open source software provides tools for data preprocessing, implementation of several Machine Learning algorithms, and visualization tools so that you can develop machine learning techniques and apply them to real-world data mining problems

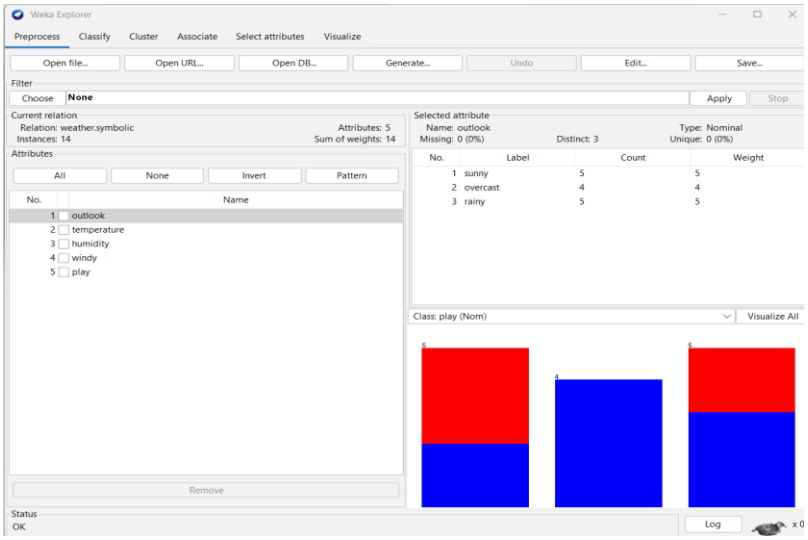
1)Loading the dataset:

- ➢ Go to weka 3.8.6

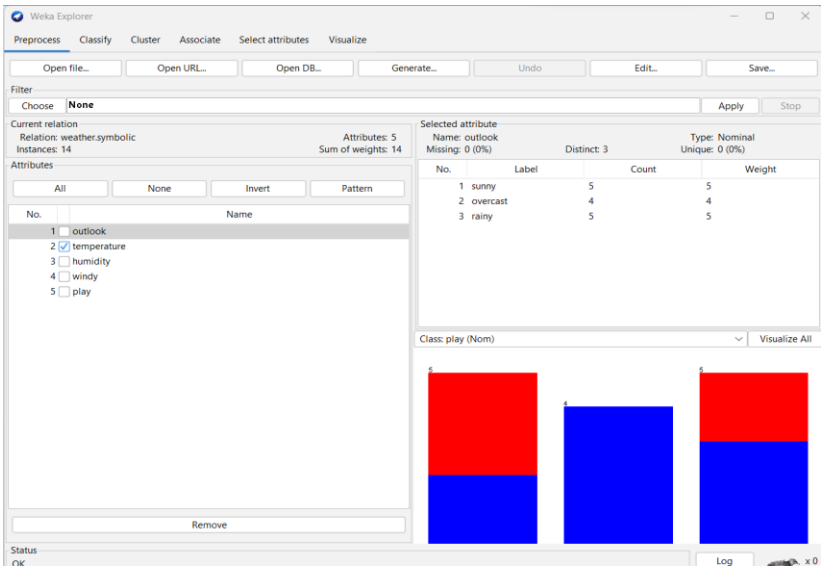
- ➢ Select "Explorer."



- ➢ Click on "Open File", then select "Look in" and navigate to Local Disk (C:) -> Program Files -> Weka-3-8-6 -> data -> weather.nominal (any data can be selected). Finally, click on "Open."
- ➢ The below data is about the “outlook data”



- Select or tick the "Temperature" box. It should appear as follows:

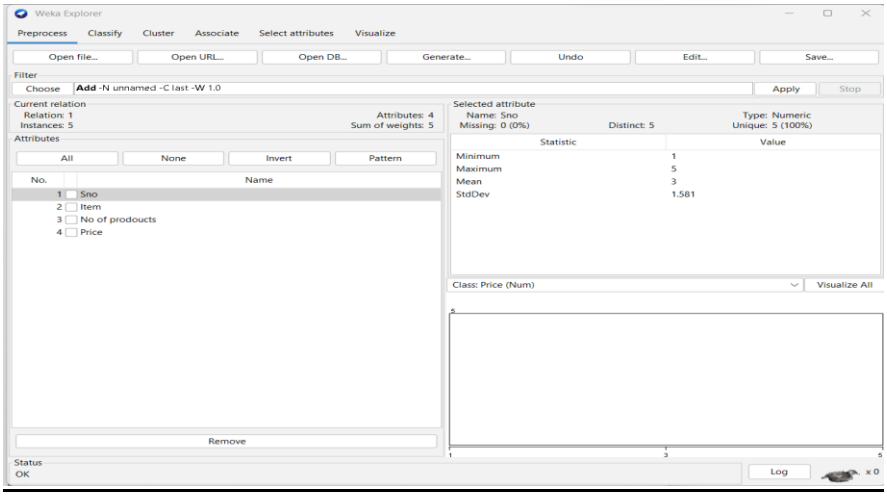


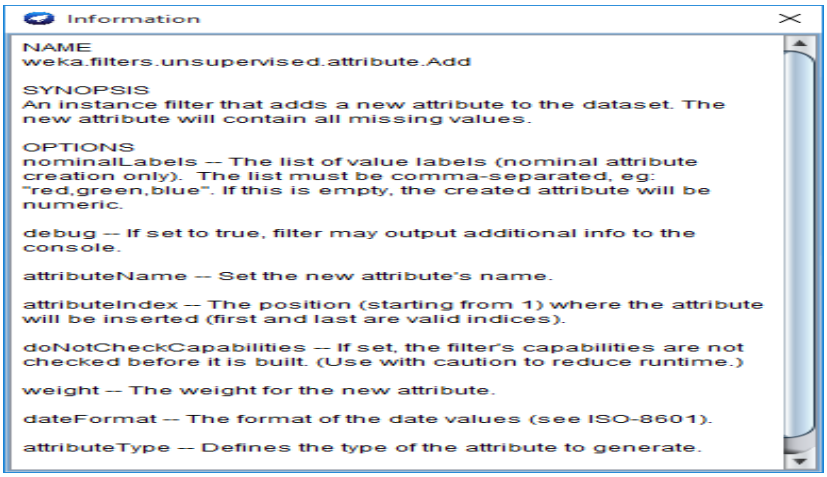
2)Adding Attribute:

- Go to excel and create the excel with some data and save it as .csv(delimiter)

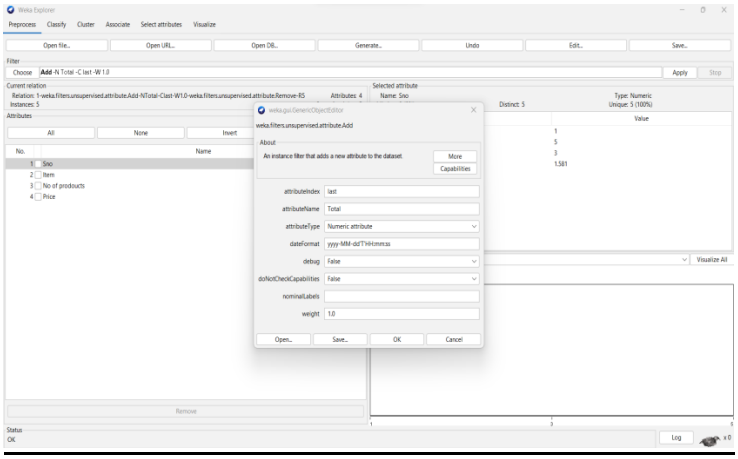
Sno	Item	No of prodoucts	Price
1	Laptop	2	800
2	Smartphone	1	300
3	Tablet	3	200
4	Headphone	1	100
5	Bluetooth	2	150

- Go to 'Open File'. This is where the Excel file is saved. Provide the file name and location. The file type should be 'csv'. Click 'Open'.

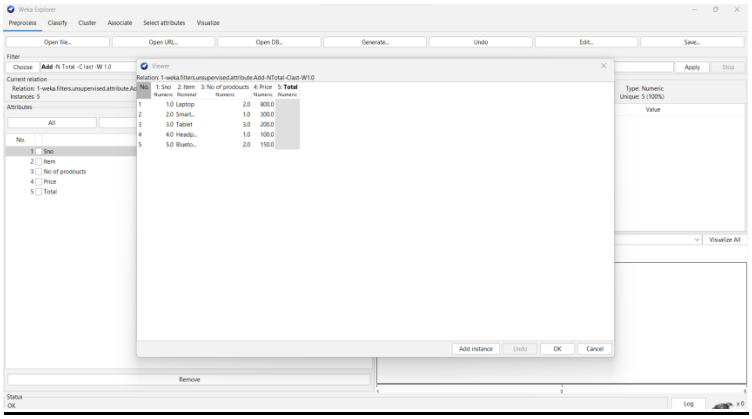




- Select choose ->filter->unsupervised->attribute->ADD and click ok.
- we have an option labeled 'Add-n'. Click on that. It will prompt you with 'Attribute Index': choose from 'Last', 'First', or 'Middle'. Next, input the 'AttributeName' which is the name of the attribute.

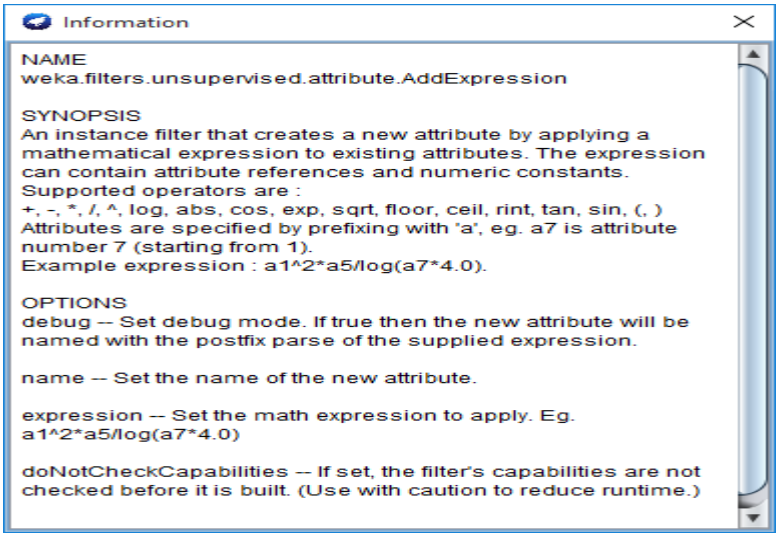


- After that, click on 'Apply' and then 'Edit'. Finally, it will display as follows

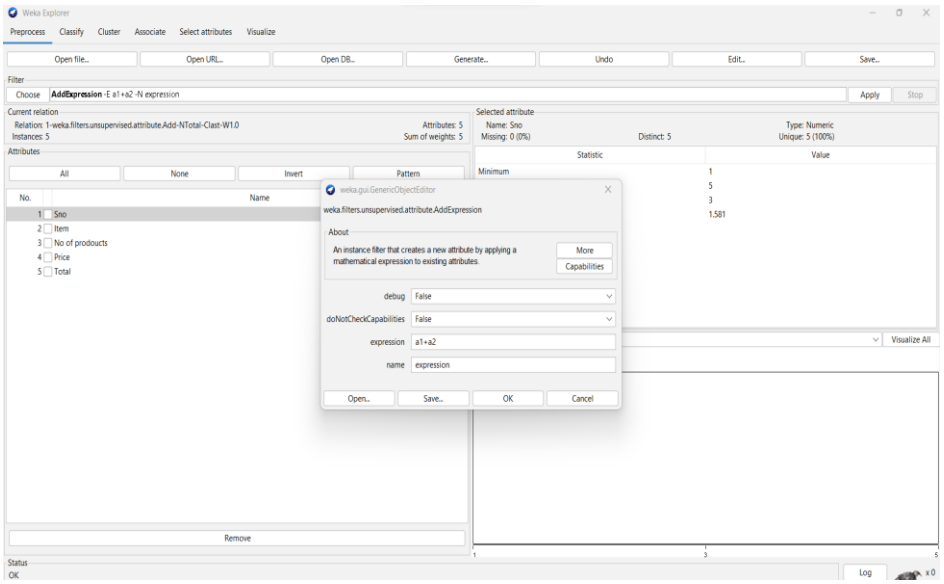


3)Add Expression:

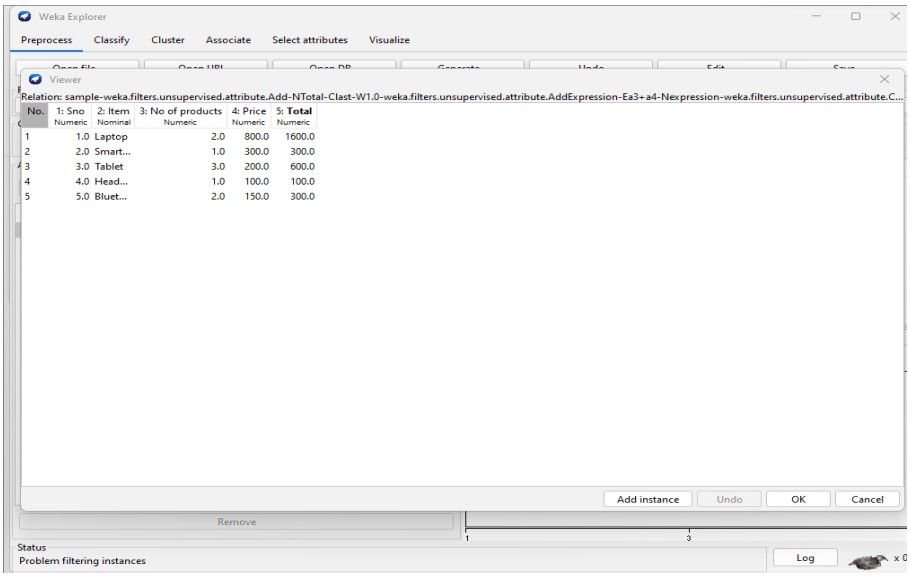
- "To add an expression, follow the same steps as adding an attribute. Select 'Choose', then navigate to 'Filter', followed by 'Unsupervised'. Next, select 'Attribute' and then choose 'Add Expression'. Click 'OK' to proceed."



- Add Expression name as"a3+a4" and Name as "Total".

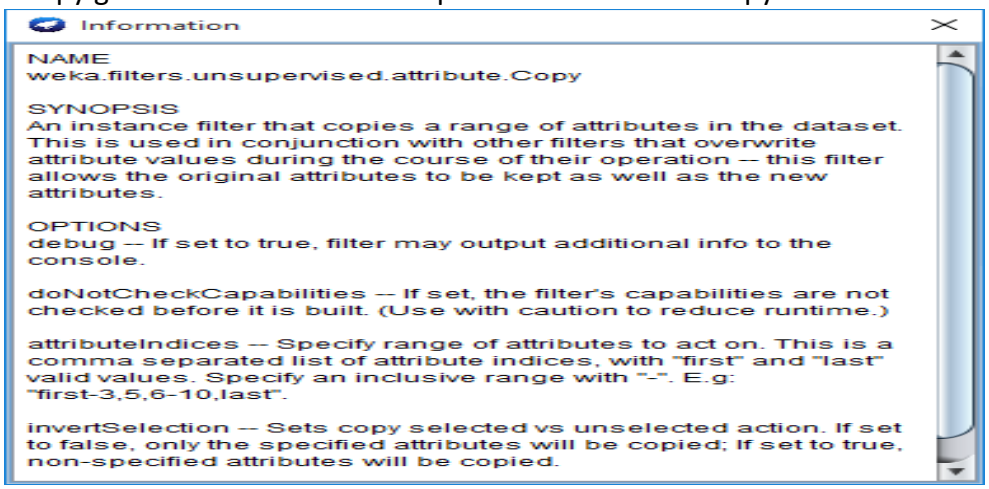


- Click the “Apply” button to apply the filter and edit

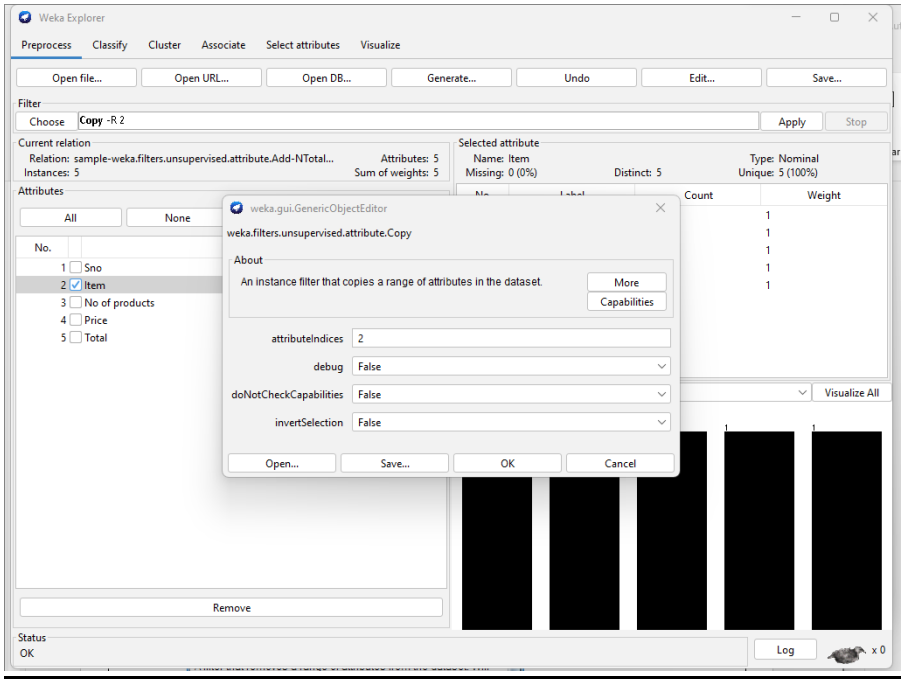


4)Copy attribute:

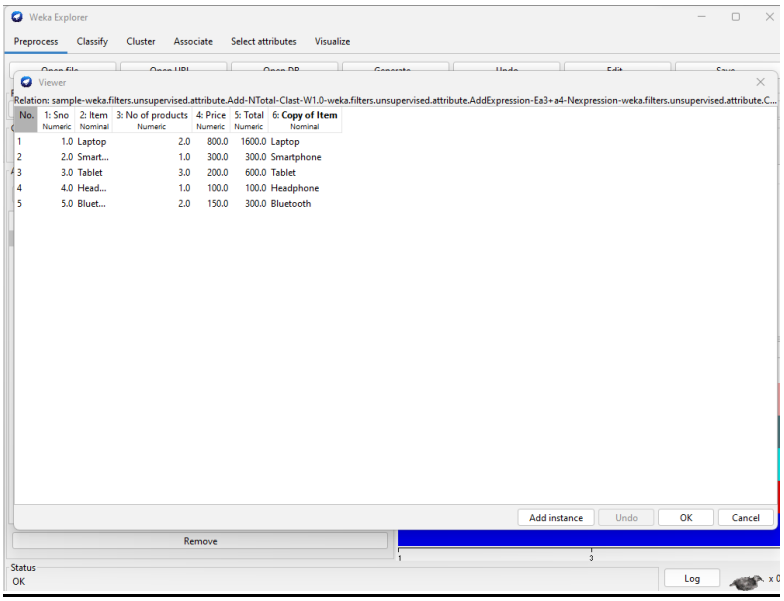
- For copy go to choose->filter->unsupervised->attribute->copy->ok



- copy attribute at last indices to the dataset

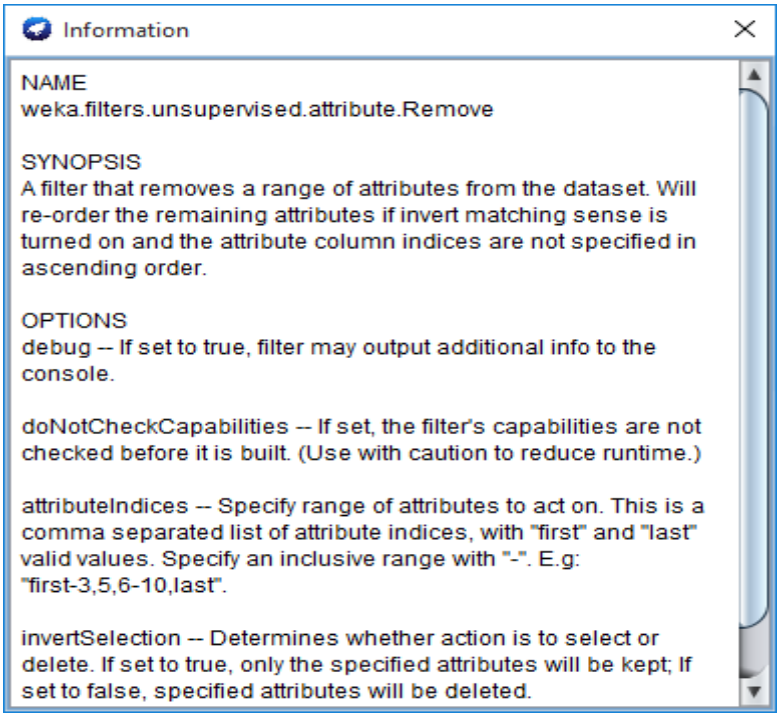


- Apply and edit

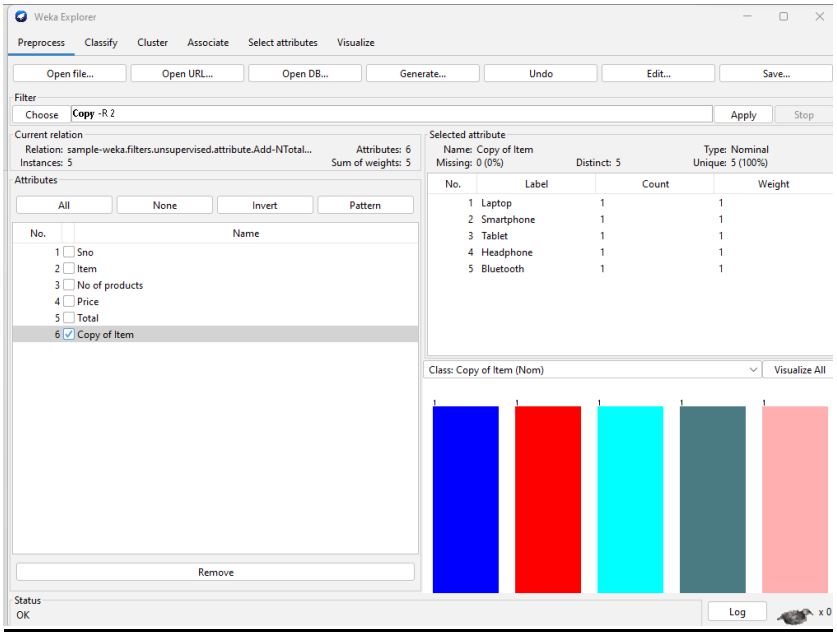


5)Remove attribute:

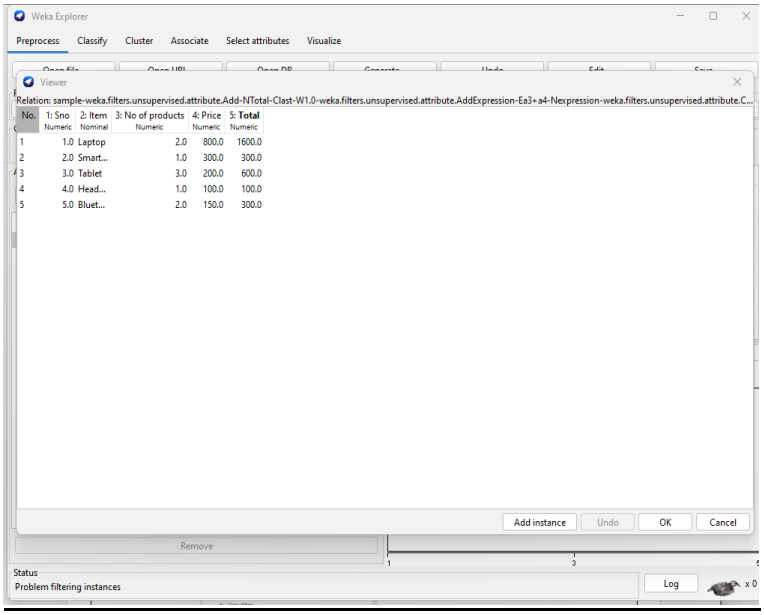
- To remove the attribute select/tick the attribute and select remove



- Select the attribute”Copy of Item” and remove the attribute(click on remove).



- Click apply and edit.


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Viewer
Relation: sample-weka.filters.unsupervised.attribute.Add-NTotal-Clast-W1.0-weka.filters.unsupervised.attribute.AddExpression-Ea3+a4-Nexpression-weka.filters.unsupervised.attribute.C...
No. 1: Sno Numeric 2: Item Nominal 3: No of products Numeric 4: Price Numeric 5: Total Numeric
1 1.0 Laptop 2.0 800.0 1600.0
2 2.0 Smart... 1.0 300.0 300.0
3 3.0 Tablet 3.0 200.0 600.0
4 4.0 Head... 1.0 100.0 100.0
5 5.0 Bluet... 2.0 150.0 300.0
Add instance
Undo
OK
Cancel
Remove
Status
Problem filtering instances
Log
x 0

EXPERIMENT – 2

AIM:

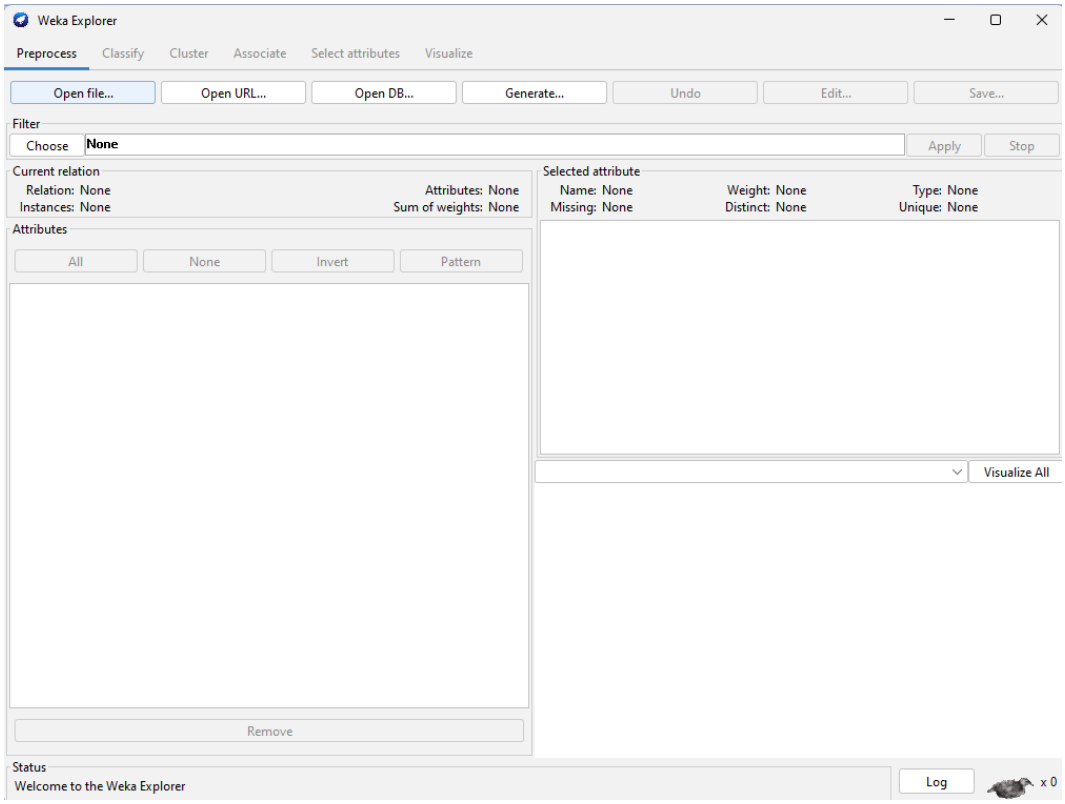
Perform operations on any dataset using ReplaceMissingValues and ReplaceMissingValues by user content and Normalize attributes.

Description:

In weka, replacing missing values is a crucial preprocessing step. Utilize the “Replace Missing Values” filter to handle missing data, selecting either default strategies like mean or median imputation or opting for user specified content replacement. This ensures a complete dataset for more accurate and reliable machine learning model training.

Procedure:

- Step-1: open weka tool and select the explorer

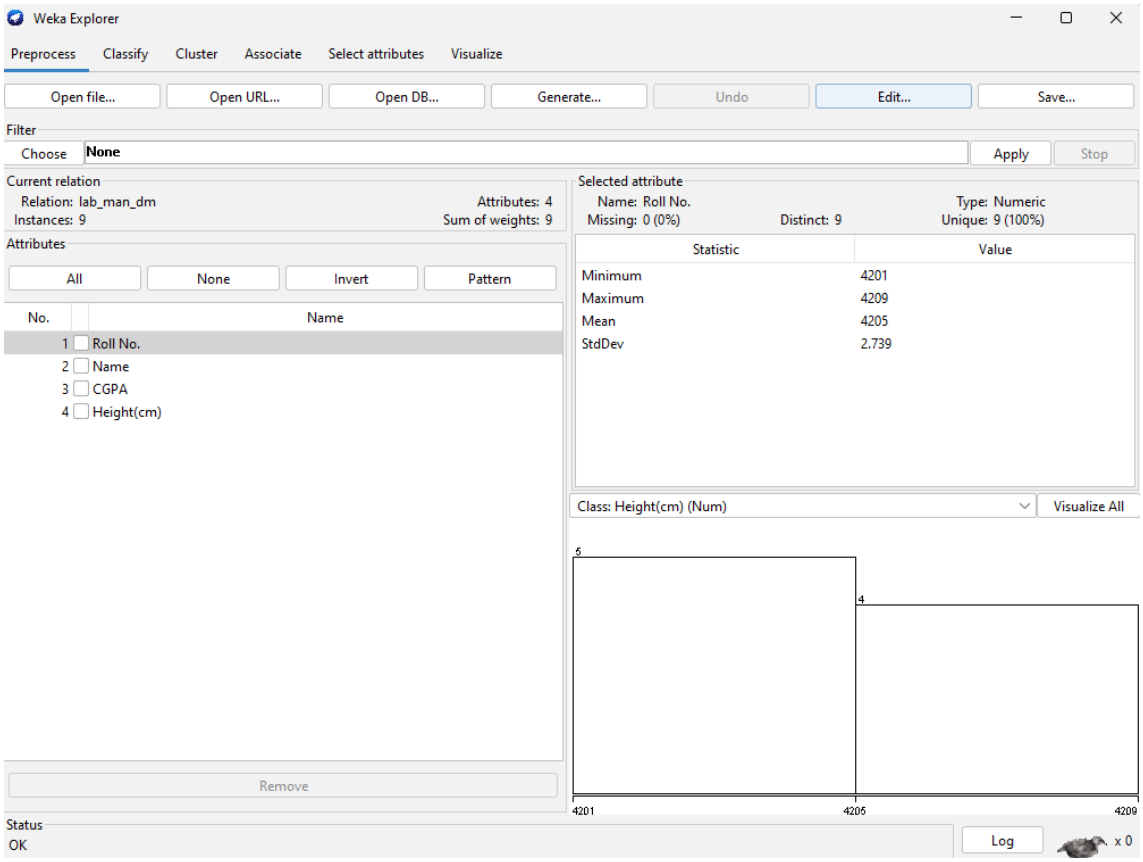


❖ Step-2: load the dataset by selecting the “open file” option.

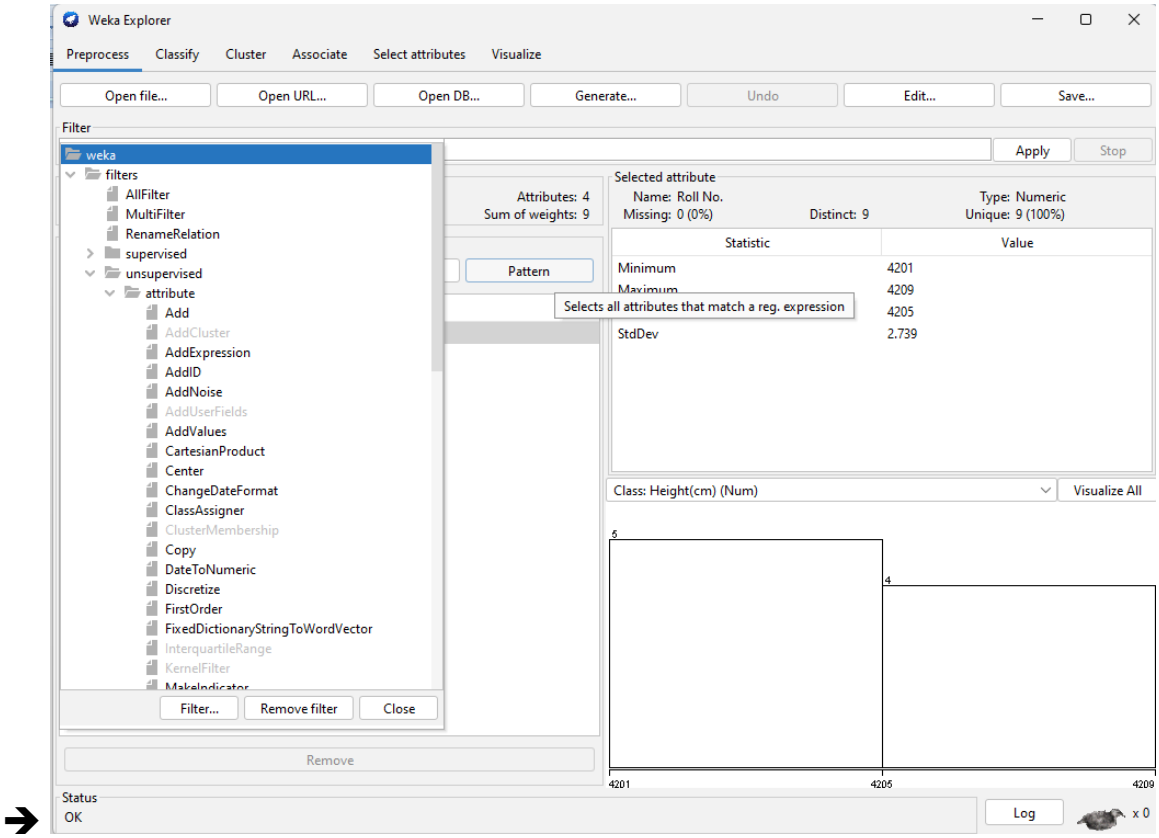
Select the “.csv(comma delimited)” saved file.

1	Roll No.	Name	CGPA	Height(cm)
2	4201	kiran kumar	9	175
3	4202	ayush	8.2	180
4	4203	manoj	8.5	165
5	4204	naveen	8.28	150
6	4205	vyshu	9.35	165
7	4206	sai	7.5	180
8	4207	kiran kumar	8	192
9	4208			
10	4209	nikhil	8.6	174

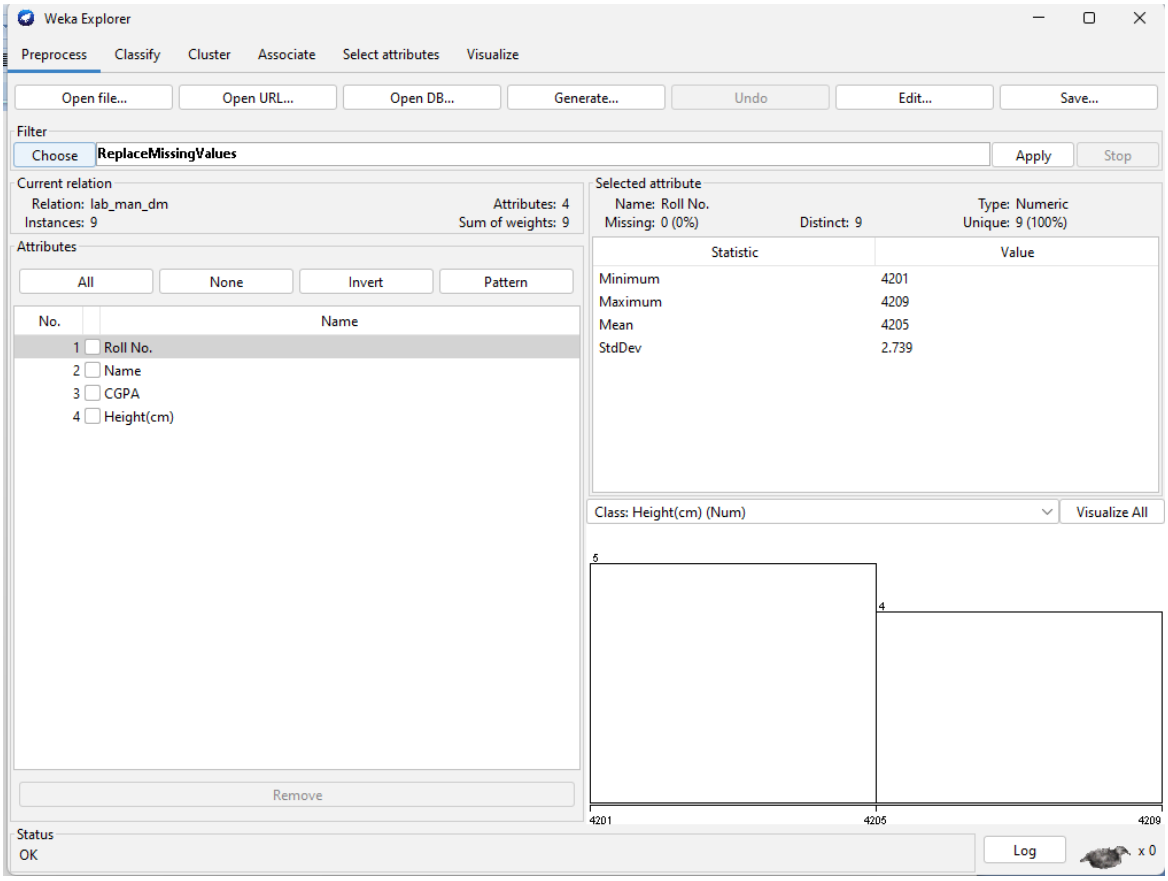
❖ Upon successful opening.



- Step-3: since the dataset is intentionally filled with missing values, we will fill it up using filters.
- ReplaceMissingValues:
  - ➔ To utilize this option, follow accordingly
  - ➔ Select choose filter > filters > unsupervised > attribute > ReplaceMissingValues

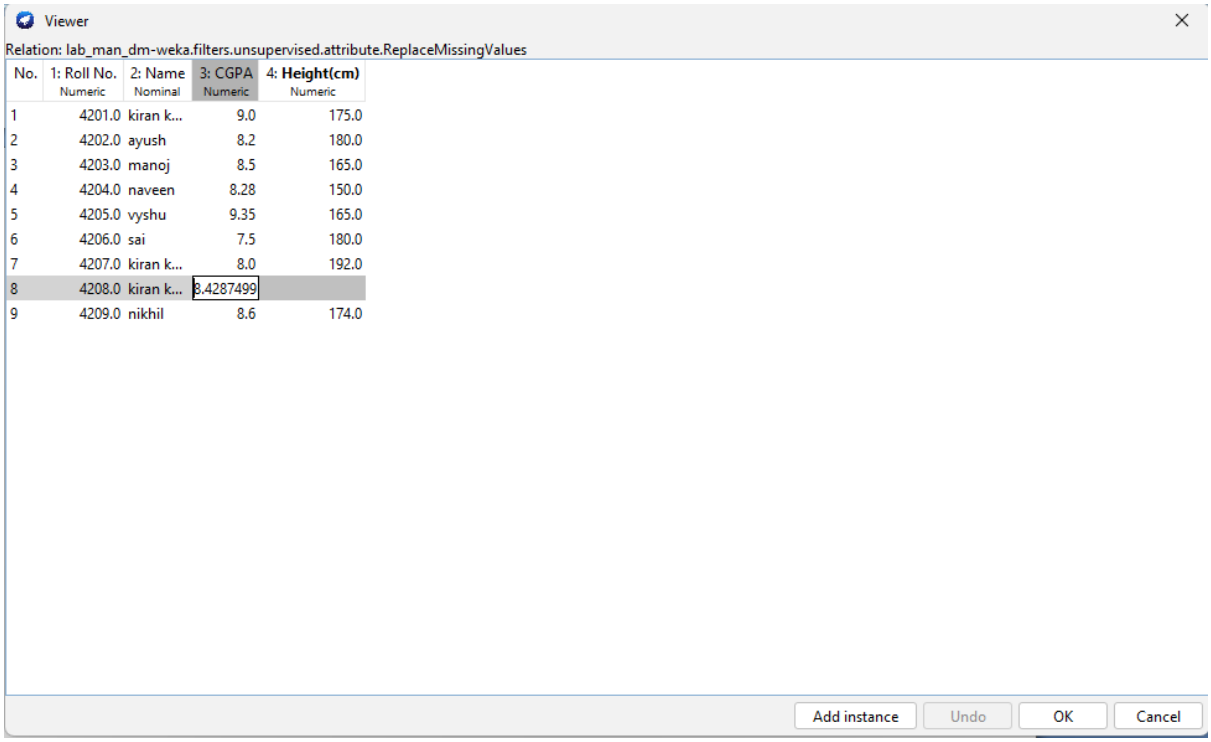


➔



➔

➔ Next, select “apply” and click on “edit” to see the result table.



➔

➔ Here, we can see that the 8th instance is filled with some values.

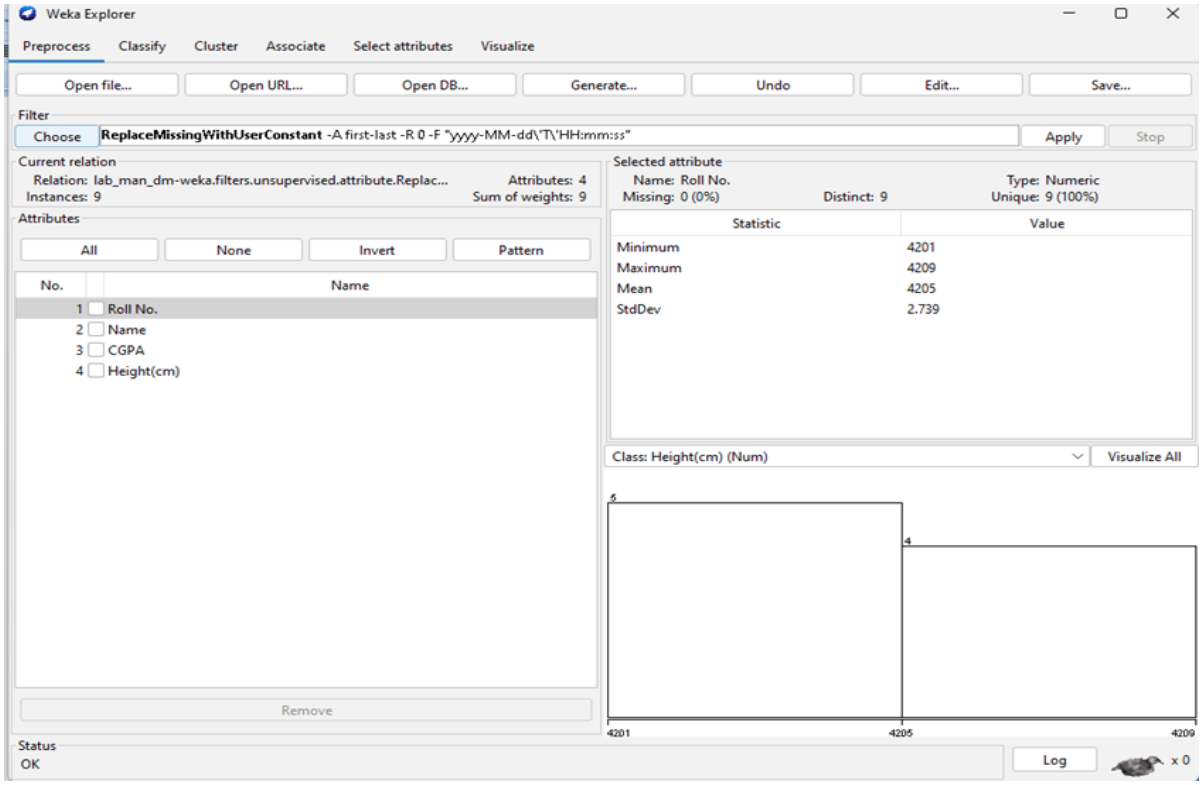
➔ NOTE:

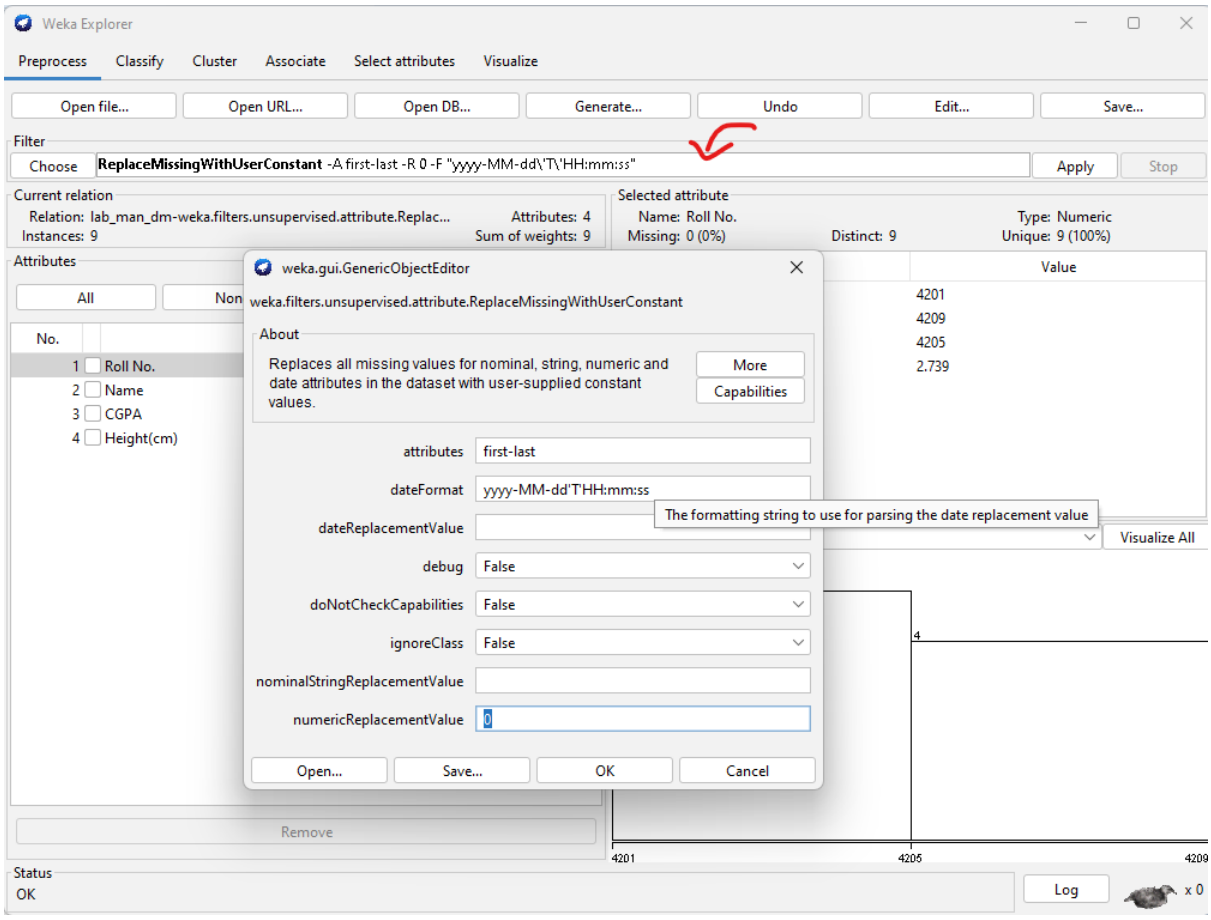
I. While replacing the Categorical data the data is replaced with Mode of the data.

II. While replacing the Numerical data the data is replaced with the Mean of the data.

III. Also, the last column is not updated because it is the decision column and no modifications are done in it.

❖ ReplaceMissingWithUserConstant:





→click below the arrow mark and modify the “numericReplacementValue” accordingly.

Viewer

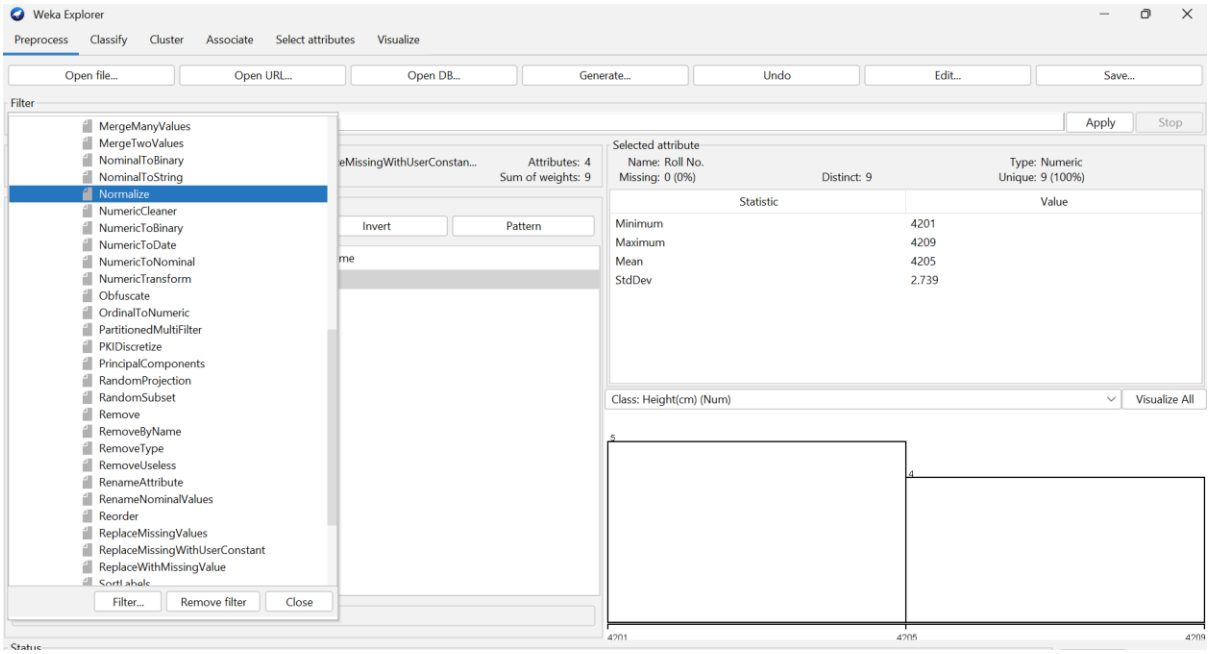
Relation: lab_man_dm-weka.filters.unsupervised.attribute.ReplaceMissingWithUserConstant

No.	1: Roll No. Numeric	2: Name Nominal	3: CGPA Numeric	4: **Height(cm)** Numeric
1	4201.0	kiran k...	9.0	175.0
2	4202.0	ayush	8.2	180.0
3	4203.0	manoj	8.5	165.0
4	4204.0	naveen	8.28	150.0
5	4205.0	vyshu	9.35	165.0
6	4206.0	sai	7.5	180.0
7	4207.0	kiran k...	8.0	192.0
8	4208.0	0.999	0.999	
9	4209.0	nikhil	8.6	174.0

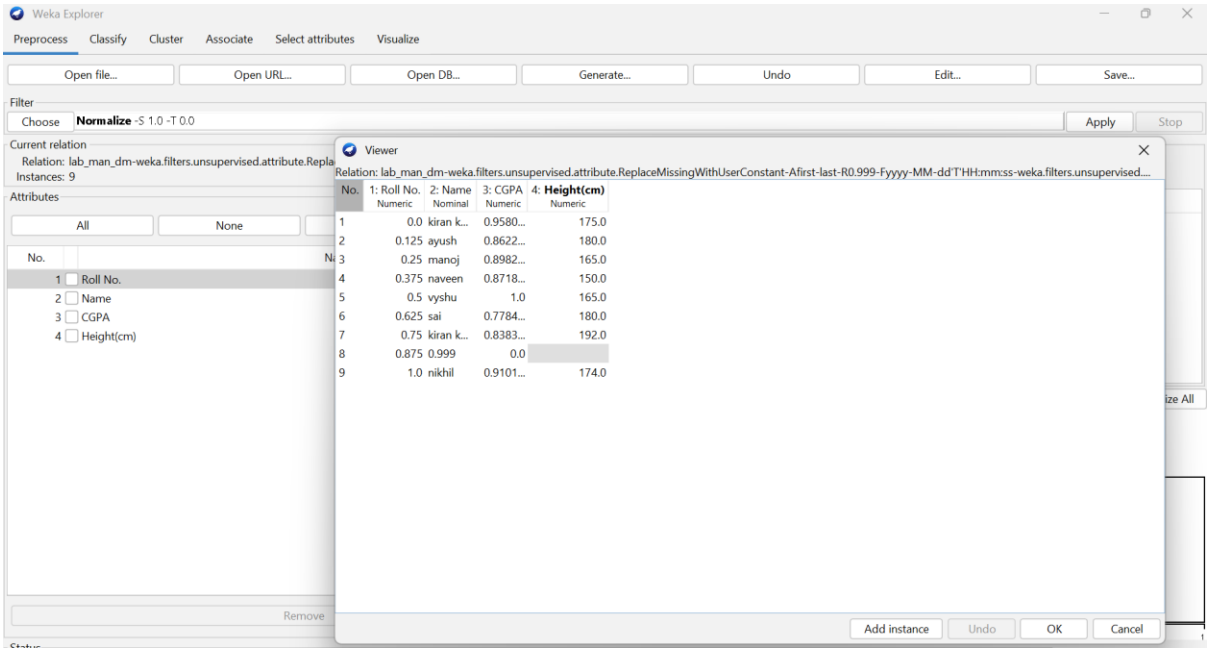
❖ Normalization:

→To utilize this option, follow accordingly

→Select choose filter > filters > unsupervised > attribute >normalize



➔ After applying the filter, results are shown below



➔ All attributes are updated to be ranged between the interval [0,1]

EXPERIMENT - 3

AIM:

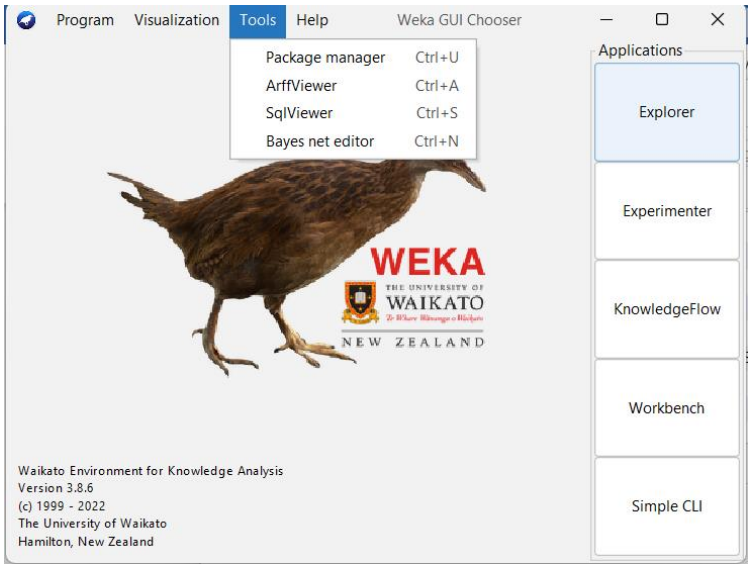
Build a Classification Model using ID3 Decision Tree Algorithm on any dataset.

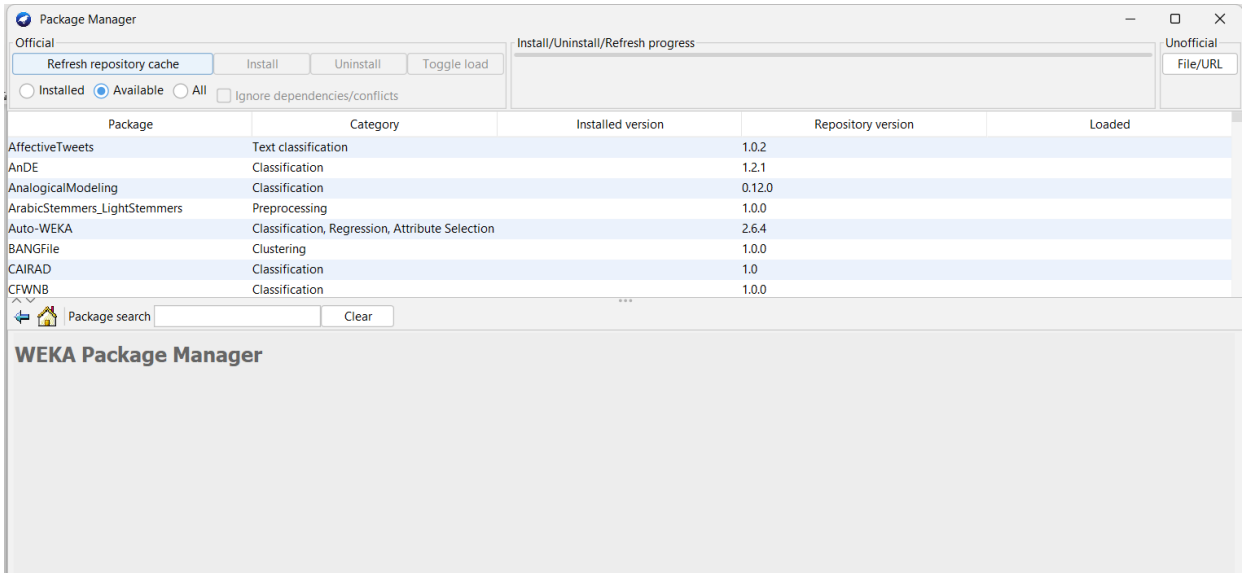
Description:

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. ID3 stands for **Iterative Dichotomiser 3** and is named such because the algorithm iteratively (repeatedly) dichotomizes (divides) features into two or more groups at each step.

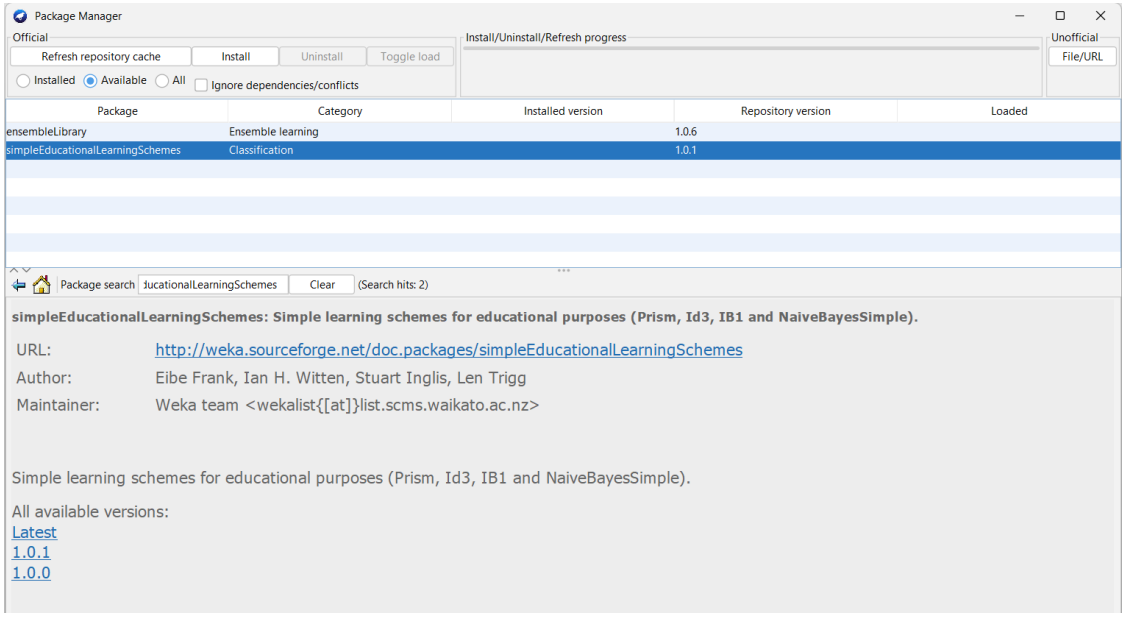
→Installing a Package

1. Select Tools > Package manager from WEKA GUI Chooser.

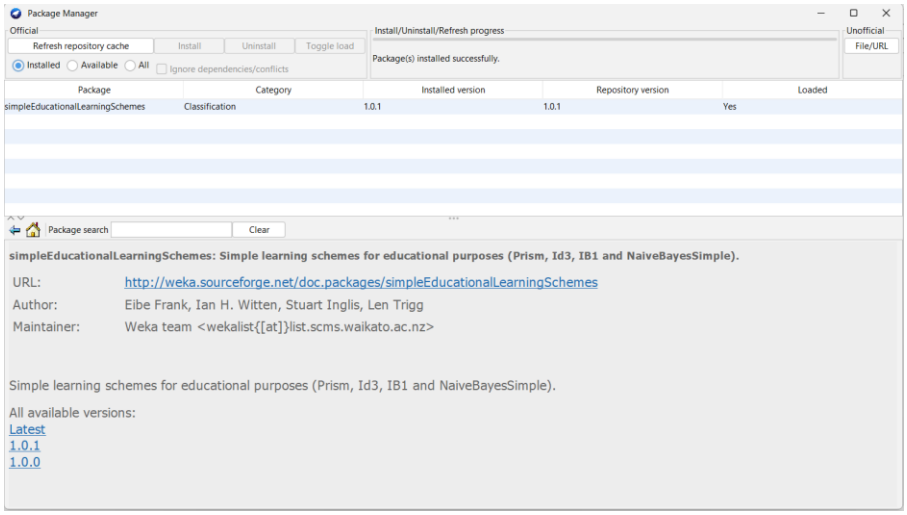




2. To download the ID3 Decision Tree package, you'll need to search for the package name "**simpleEducationalLearningSchemes**" in the Package Search. Make sure to keep the window in the "Available" mode while searching.
3. Please click the "install" button to proceed with package installation.



4. The installation process has been completed. Please check the "Installed" window to verify the package.

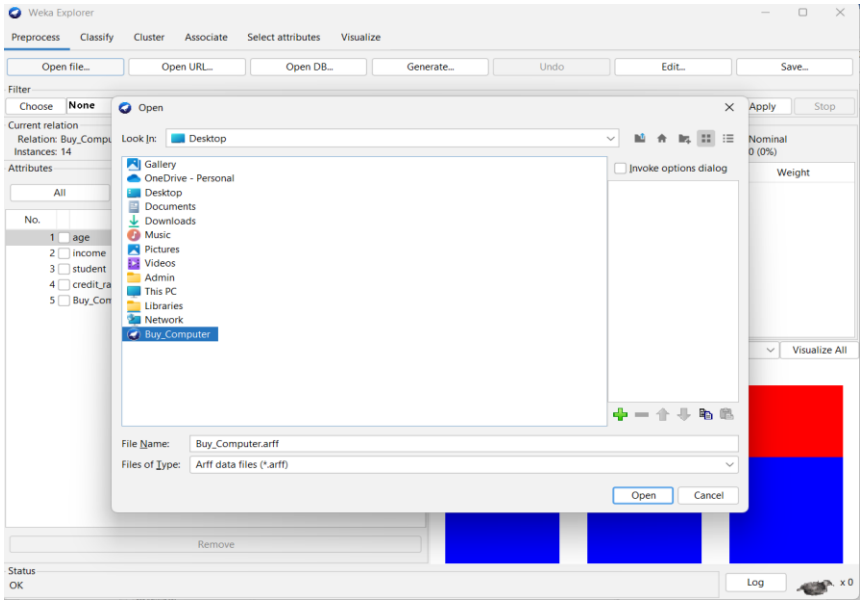


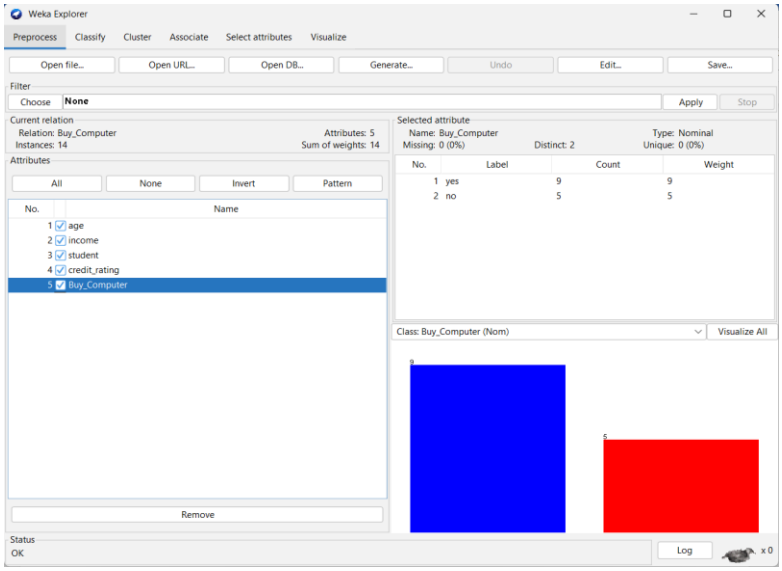
→Classification Model using ID3 Decision Tree Algorithm.

1. Generate a dataset containing attributes such as Age, Income, Student, Credit_Rating, and Buy_Computer, formatted in Notepad with .arff file format.

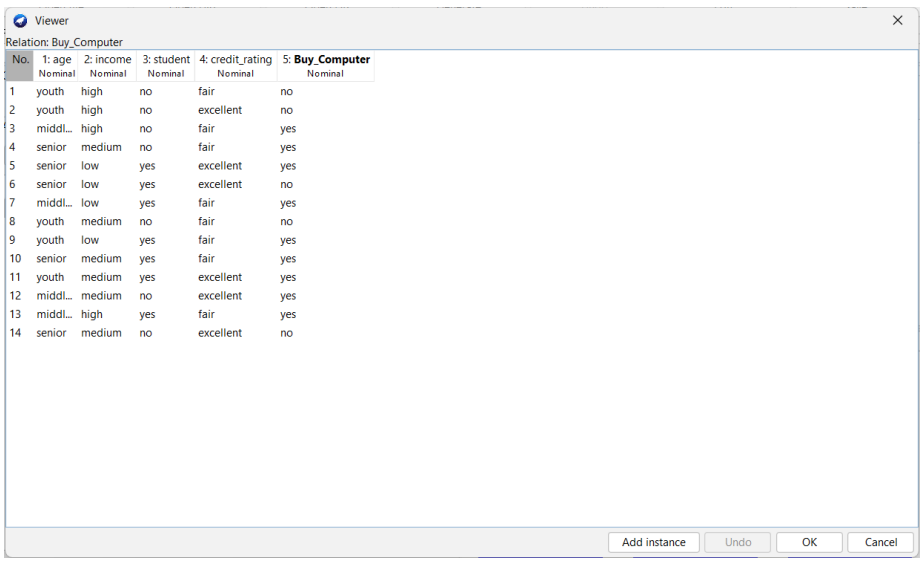
```
@relation Buy_Computer
@attribute age {youth, middle_age, senior}
@attribute income {low, medium, high}
@attribute student {yes, no}
@attribute credit_rating {excellent, fair}
@attribute Buy_Computer {yes, no}

@data
youth,high,no,fair,no
youth,high,no,excellent,no
middle_age,high,no,fair,yes
senior,medium,no,fair,yes
senior,low,yes,excellent,yes
senior,low,yes,excellent,no
middle_age,low,yes,fair,yes
youth,medium,no,fair,no
youth,low,yes,fair,yes
senior,medium,yes,fair,yes
youth,medium,yes,excellent,yes
middle_age,medium,no,excellent,yes
middle_age,high,yes,fair,yes
senior,medium,no,excellent,no
```

2. Select the dataset that we created earlier.

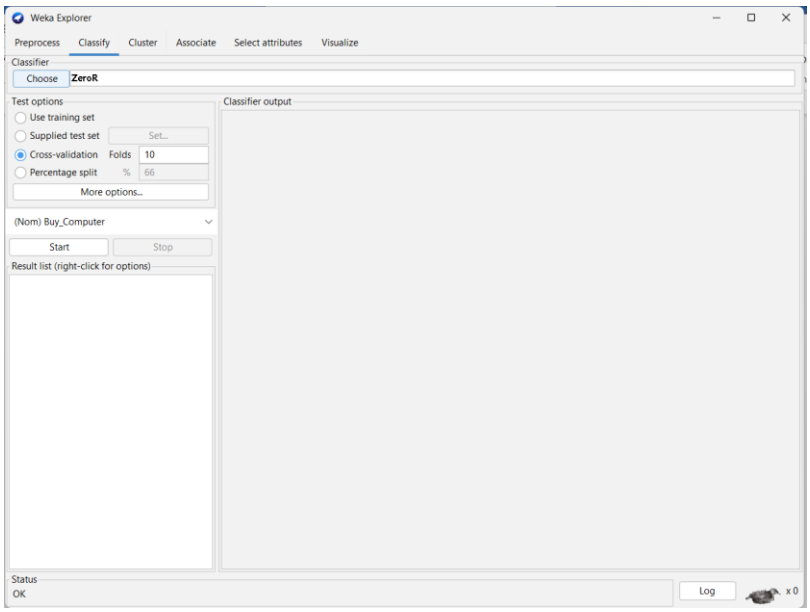




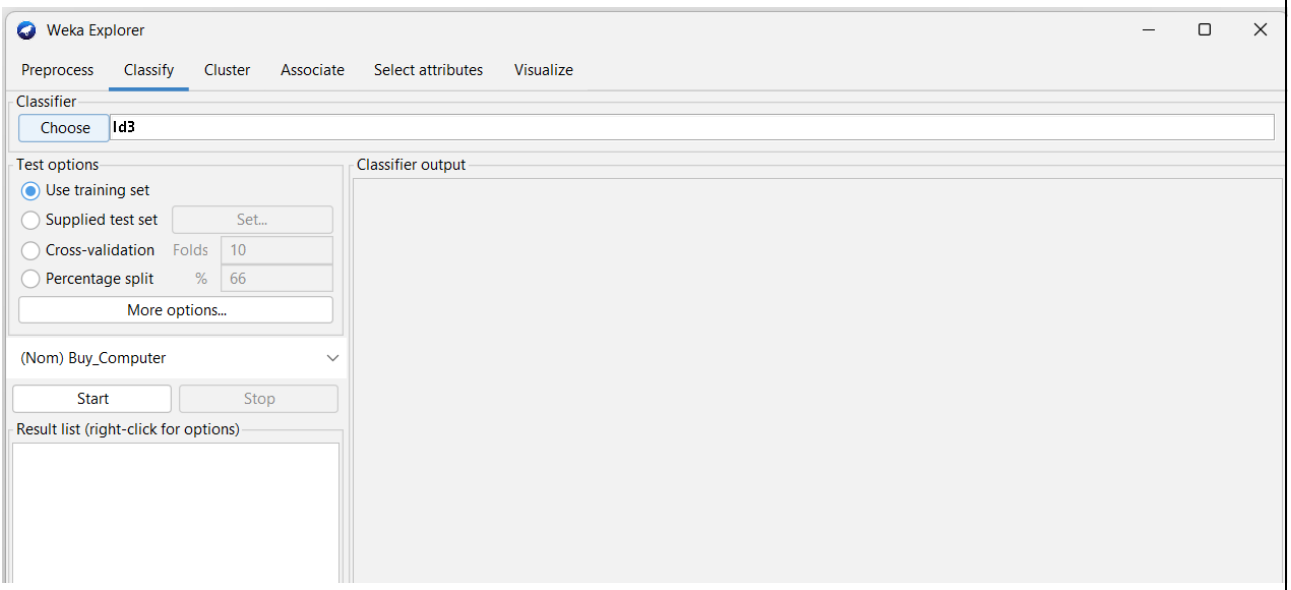
3. Please click "Edit" to view the Dataset in tabular format.



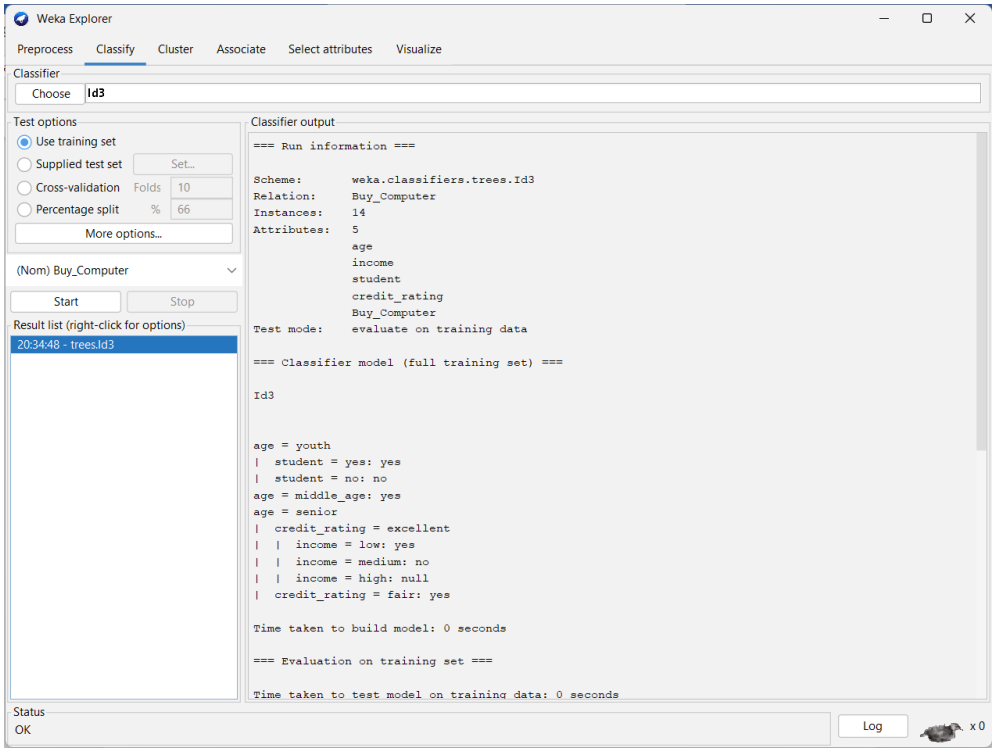
4. Please open the Classify window to utilize the **ID3 Decision Tree algorithm** for classifying the dataset.



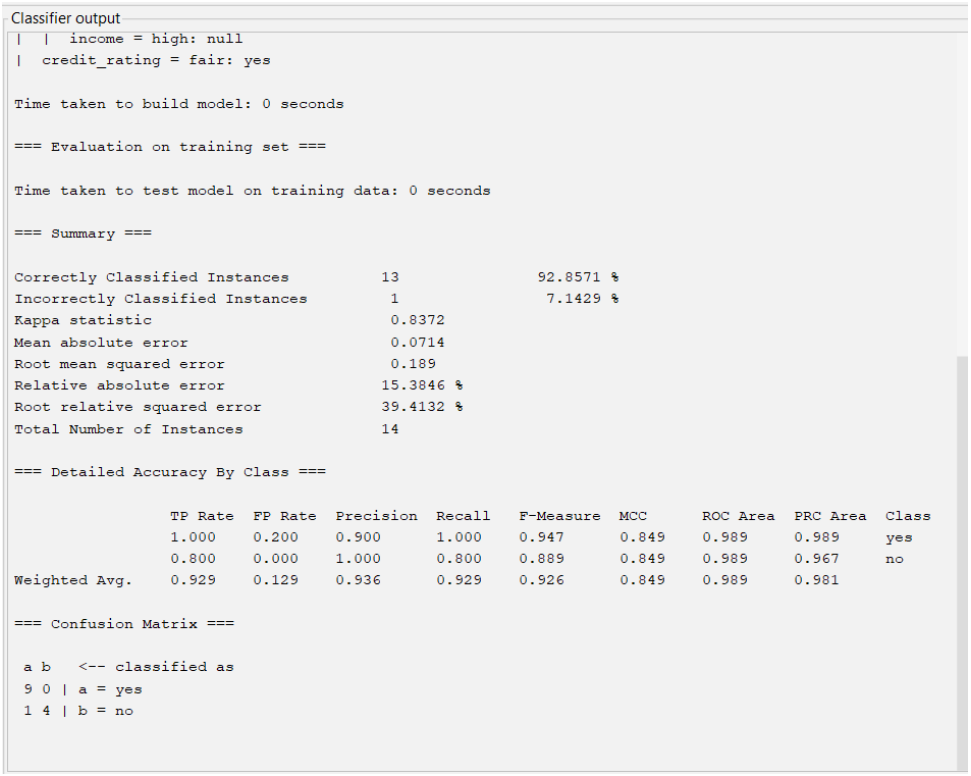
5. Click on "Choose" and select "trees > ID3" to choose the package for the ID3 Decision Tree algorithm.



6. Initiate the classification process by clicking on the "Start" button.



7. The Classifier Output window will display the following information:
    - Run information
    - Classifier model
    - Stratified cross-validation results
    - Detailed Accuracy By Class
    - Confusion Matrix



Experiment – 4

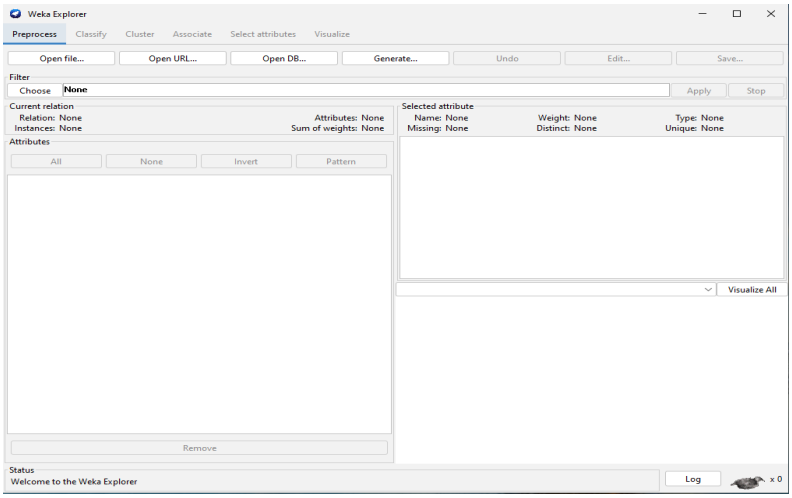
Aim:

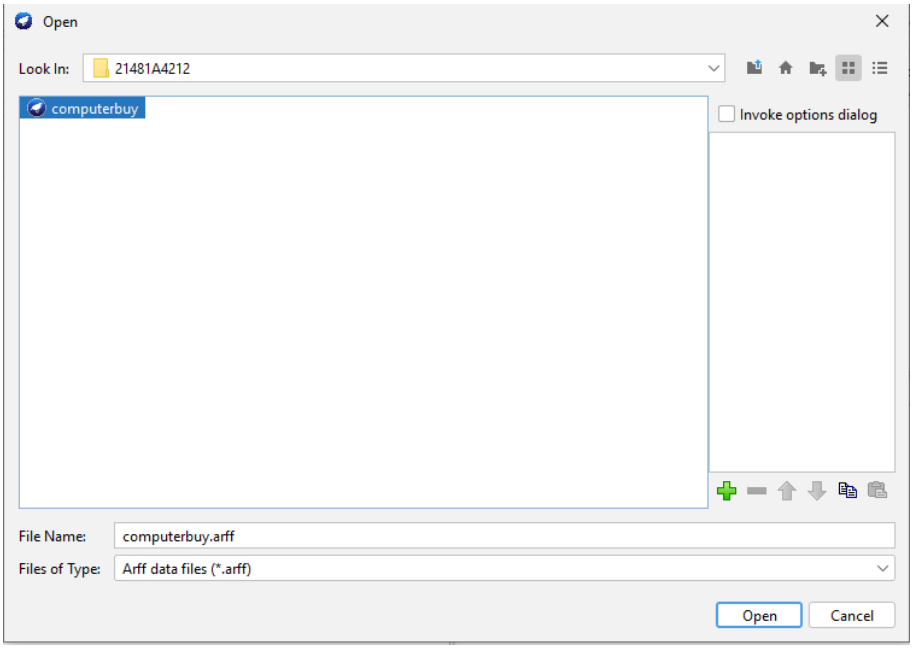
Build a Classification Model using J48 Decision Tree Algorithm on any dataset.

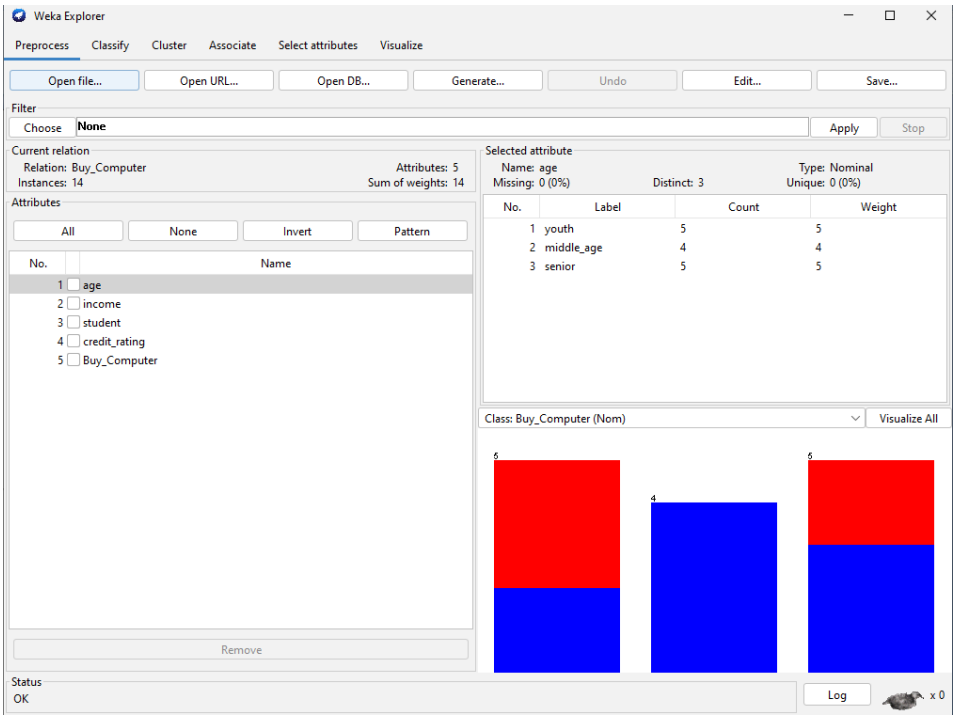
Description: J48 is a machine learning decision tree classification algorithm based on Iterative Dichotomiser 3. It is very helpful in examine the data categorically and continuously.

Process:

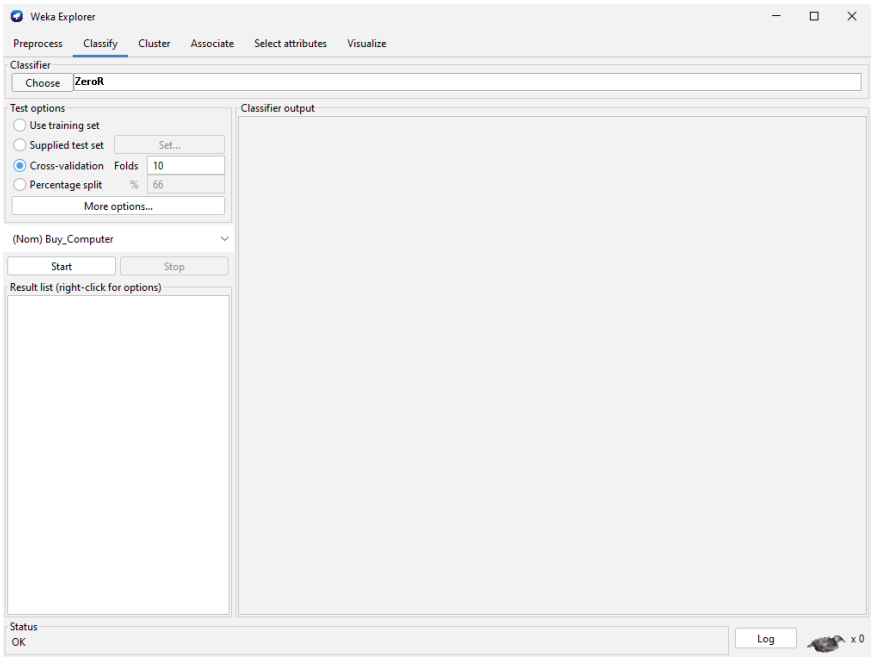
1)open weka tool > explorer and load the .arrf file

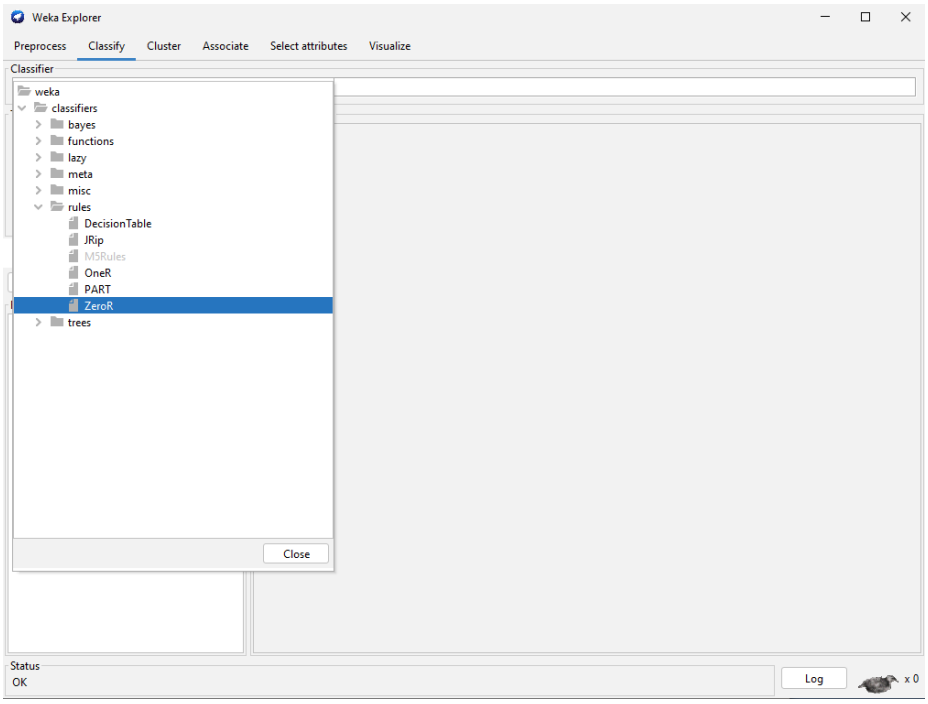


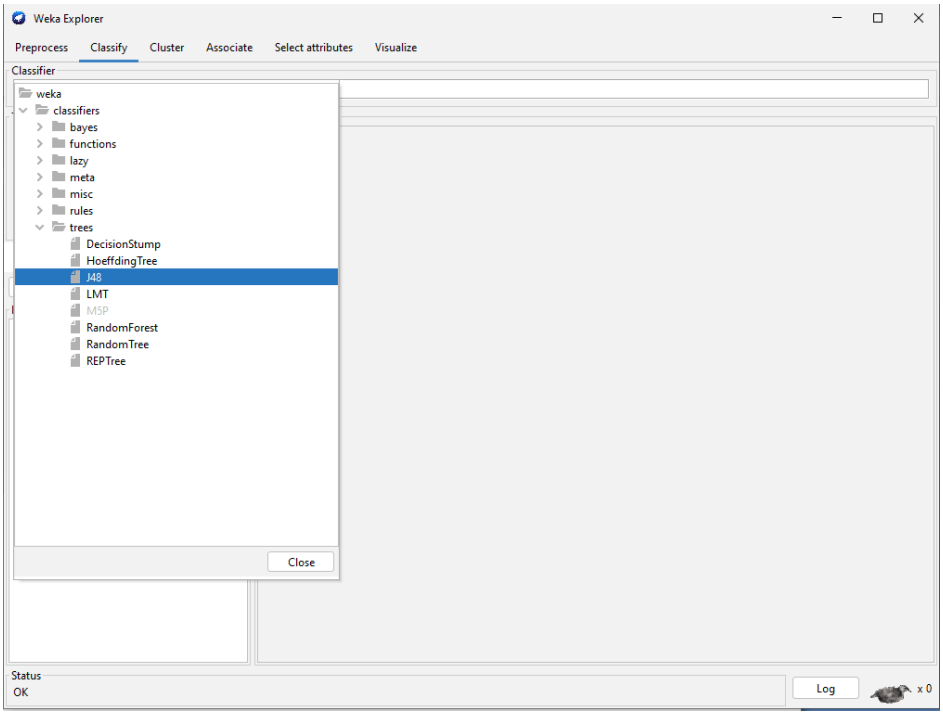




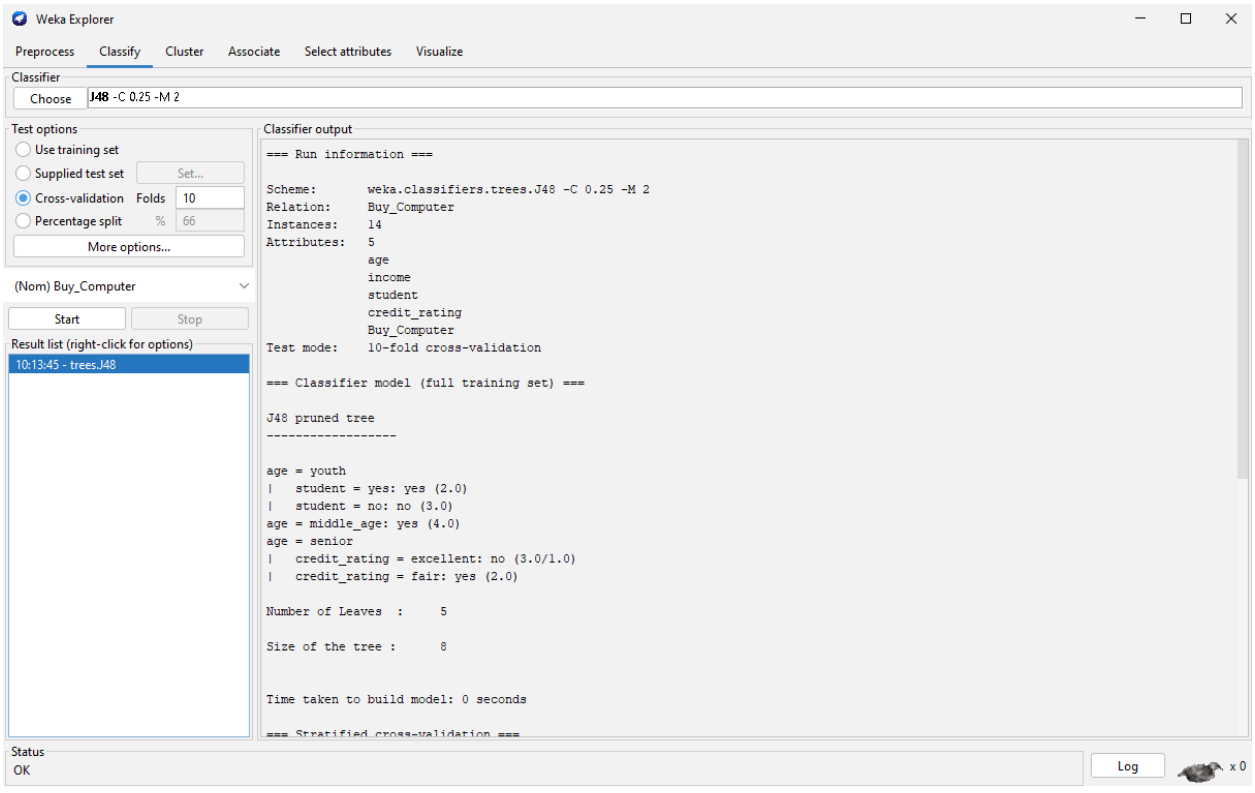
2)click on classify > choose> trees > and select J48.

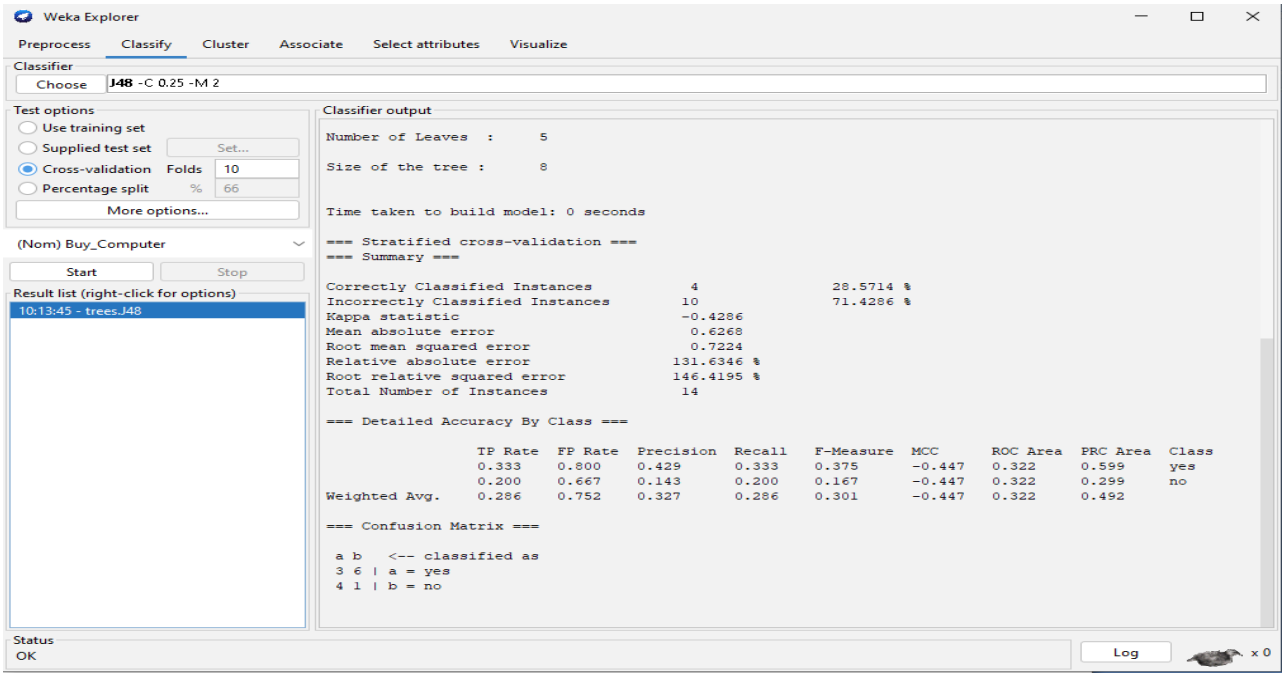

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
ZeroR
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
10
Percentage split
%
66
More options...
Classifier output
(Nom) Buy_Computer
Start
Stop
Result list (right-click for options)
Status
OK
Log
x 0


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
weka
classifiers
bayes
functions
lazy
meta
misc
rules
DecisionTable
JRip
M5Rules
OneR
PART
ZeroR
trees
Close
Status
OK
Log
x 0



3)click start and the output is observed in screen.

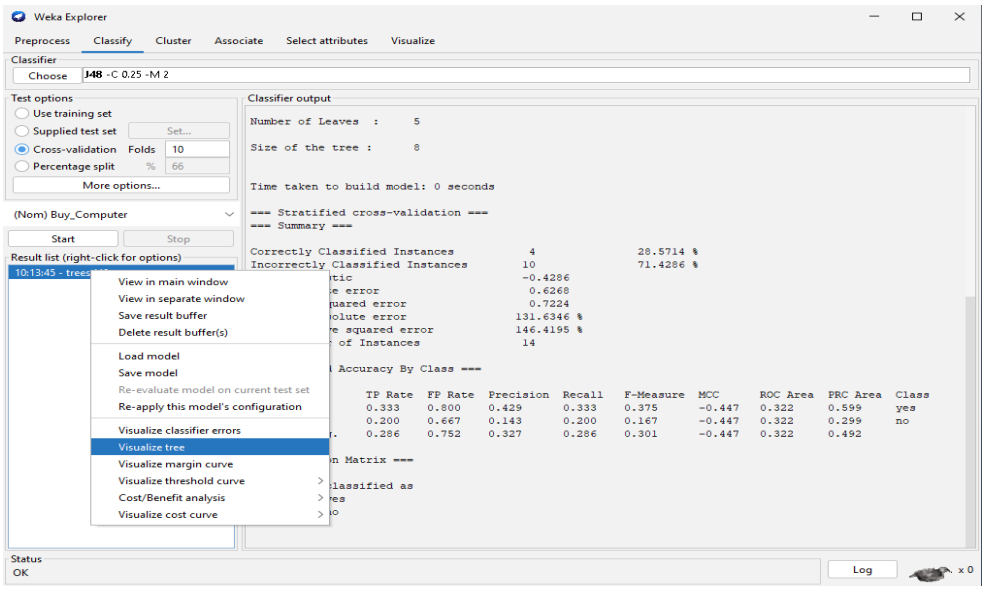


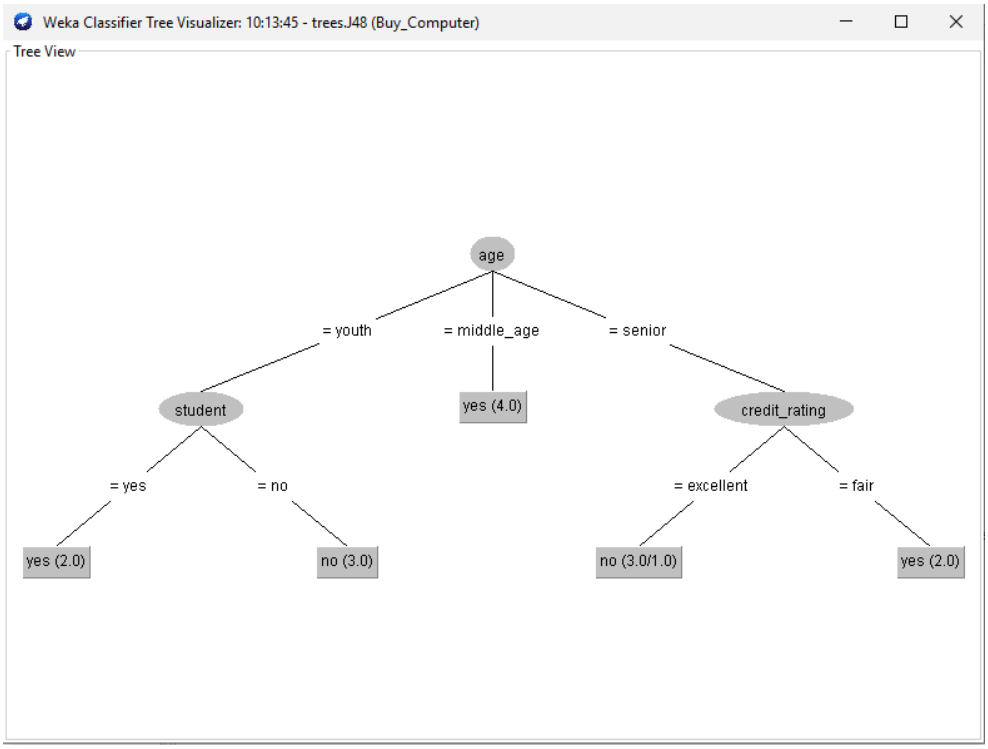


4) The classifier output window will display the following information

I. Run Information
II. Classifier model
III. Stratified cross –validation results
IV. Detailed Accuracy by class
V. Confusion matrix

5) To get the decision tree move cursor to the highlighted line and right click it. Now select visualise tree form the given options.




Weka Classifier Tree Visualizer: 10:13:45 - trees.J48 (Buy_Computer)
Tree View
age
= youth
= middle_age
= senior
student
yes (4.0)
credit_rating
= yes
= no
= excellent
= fair
yes (2.0)
no (3.0)
no (3.0/1.0)
yes (2.0)

Experiment -5

Aim:

Build a Classification Model using Naive Bayes Algorithm on any dataset

Description:

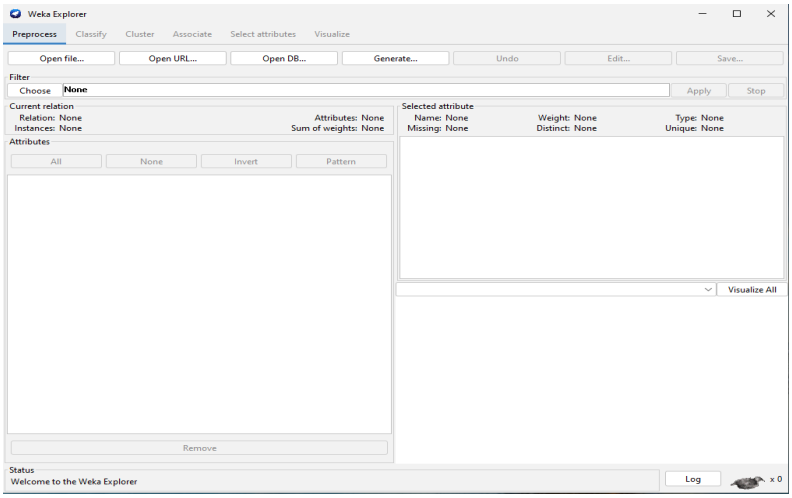
A naive Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label y.
The conditional independence assumption can be formally stated as follows:

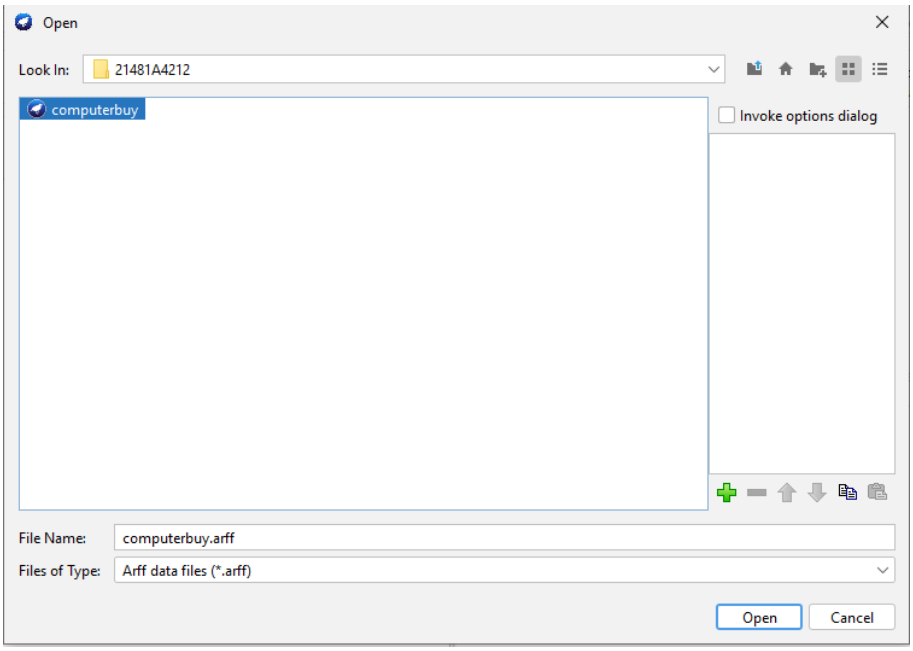
$$P(X/Y=y) = \prod_{i=1}^{d} p(\frac{Xi}{Y} = y)$$

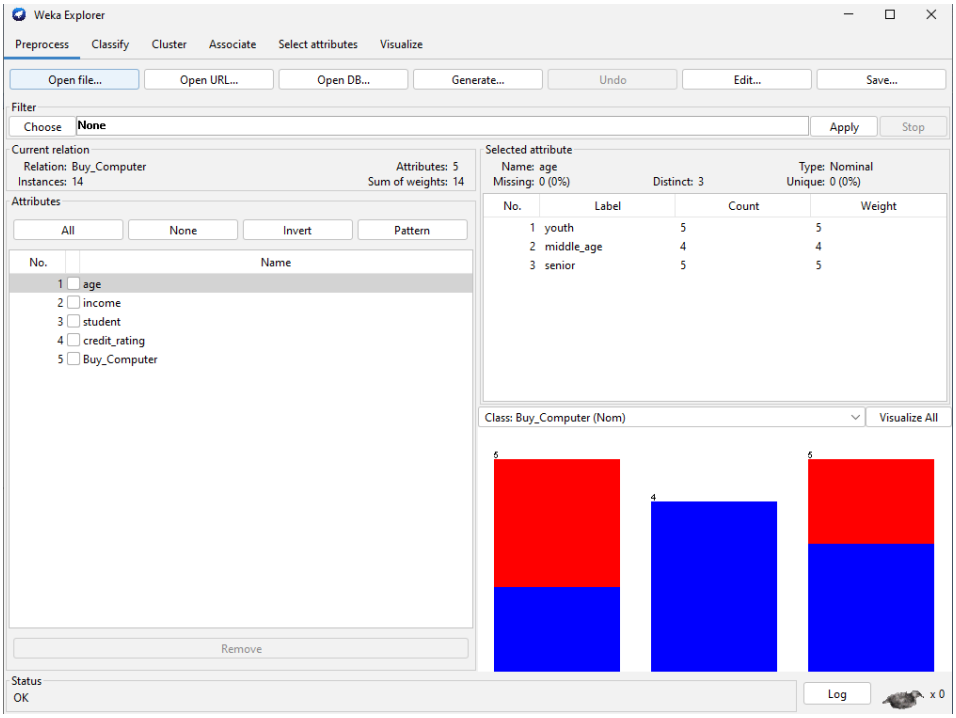
Where each attribute set $X = \{X_1, X_2,...,X_d\}$ consists of d attributes.

Process:

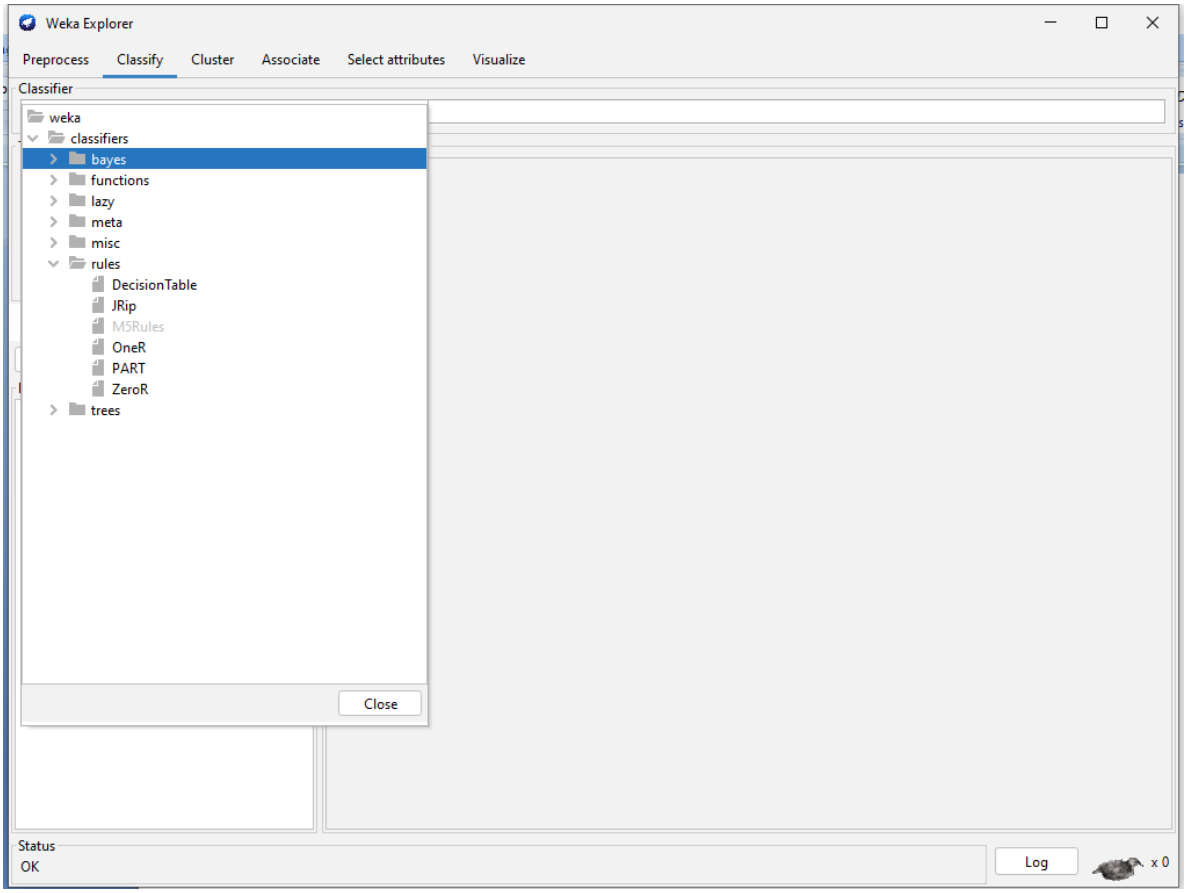
1. open weka tool > explorer and load the .arrf file

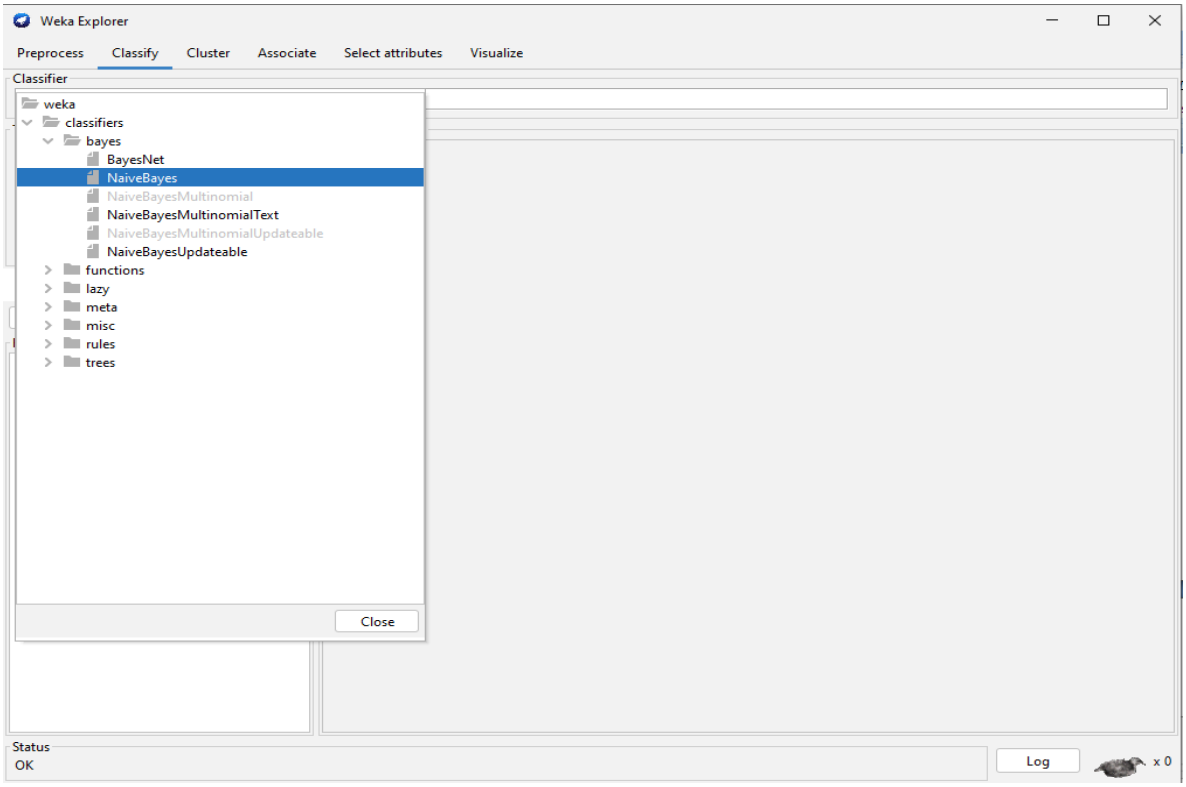

Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: None
Instances: None
Attributes: None
Sum of weights: None
Selected attribute
Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None
Attributes
All
None
Invert
Pattern
Visualize All
Remove
Status
Welcome to the Weka Explorer
Log
x 0


Open
Look In:
21481A4212
computerbuy
Invoke options dialog
File Name:
computerbuy.arff
Files of Type:
Arff data files (*.arff)
Open
Cancel

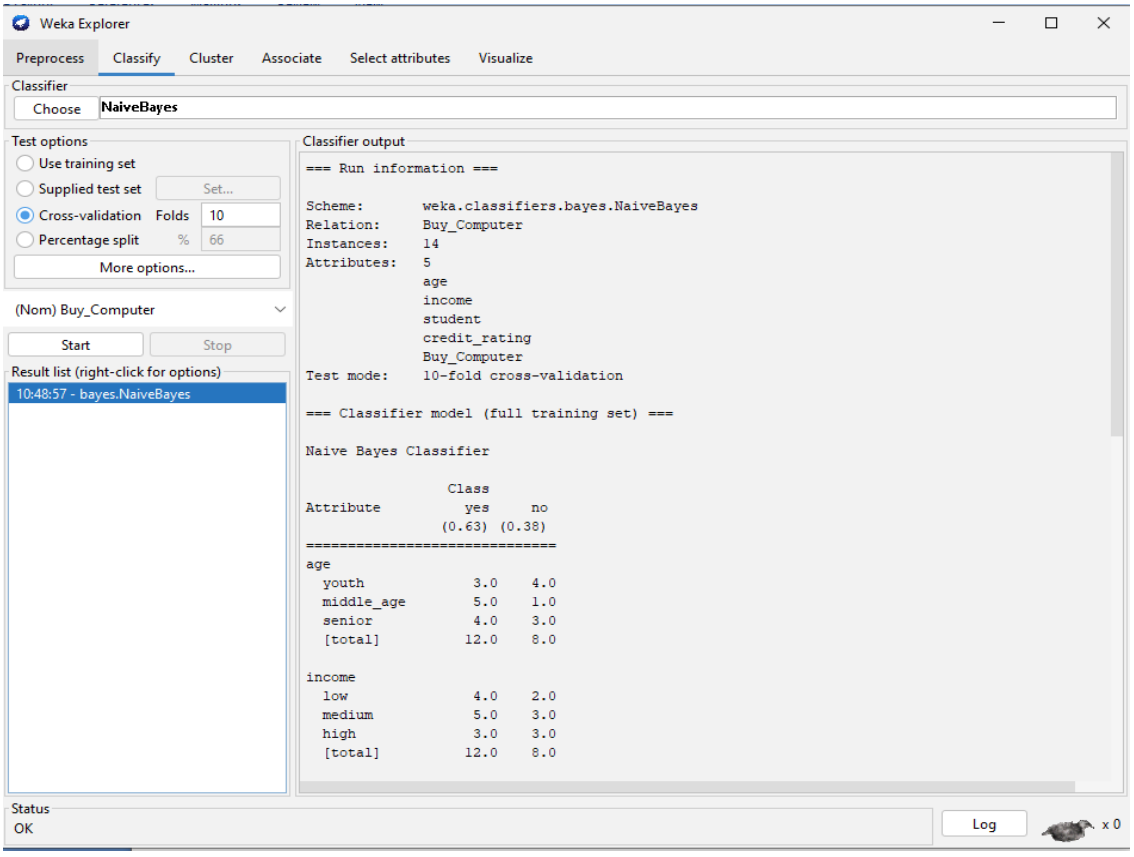


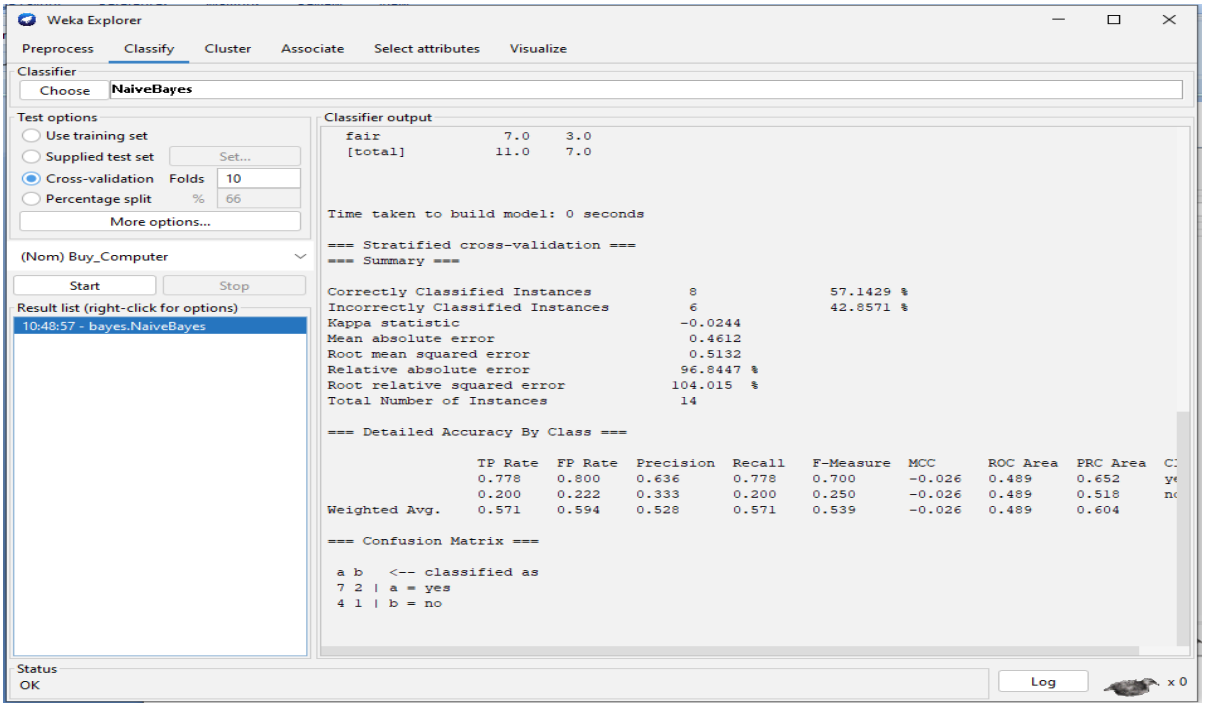
2. Click on classify > choose>bayes> and select naviebayes

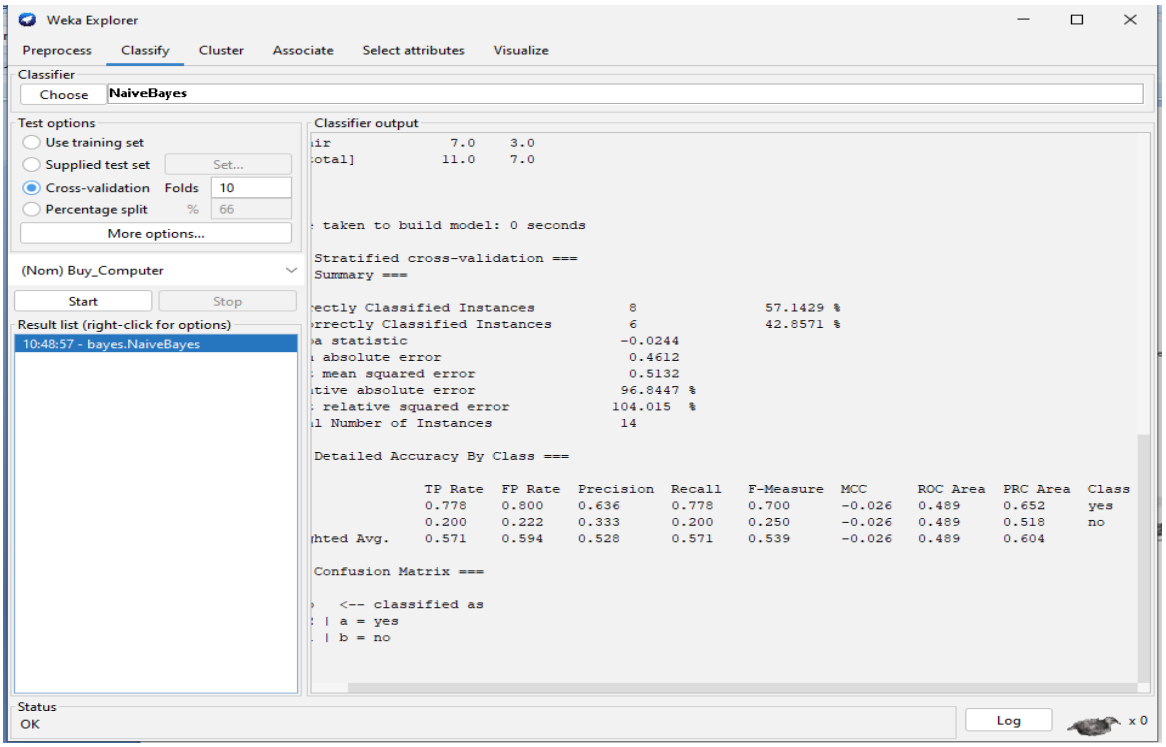




3. Click start and the output is observed in screen.







4. The classifier output window will display the following information

VI. Run Information
VII. Classifier model
VIII. Stratified cross –validation results
IX. Detailed Accuracy by class
X. Confusion matrix

EXPERIMENT - 6

AIM:

Apply K-Cross Validation on any dataset.

DESCRIPTION:

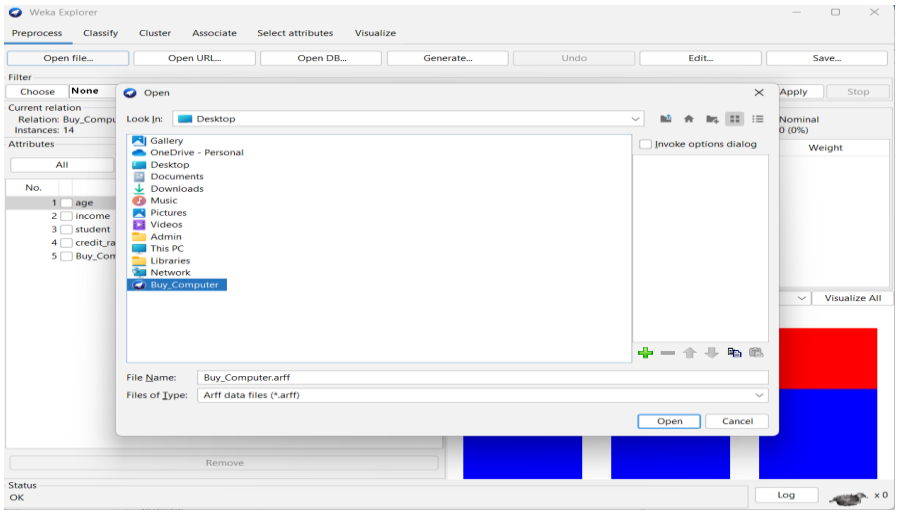
To apply K-Cross Validation as previously done in Experiment 3, we need to download the ID3 package.

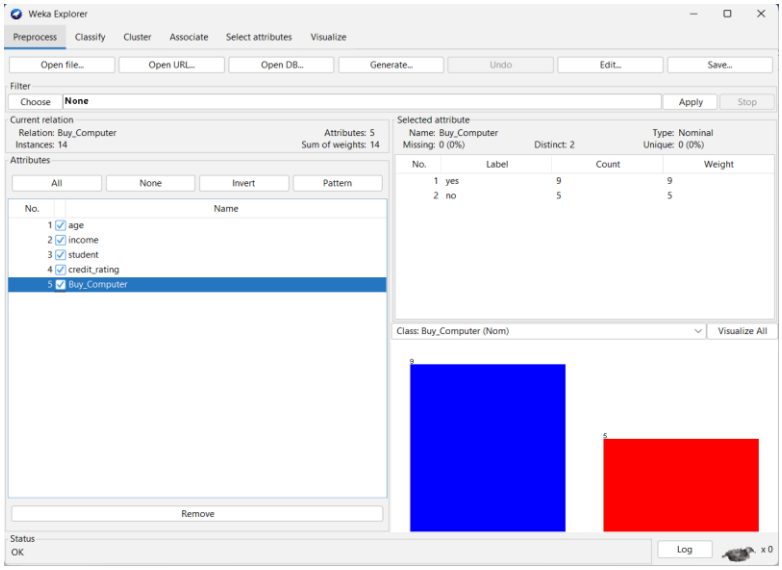
→Apply K-Cross Validation

1. Generate a dataset containing attributes such as Age, Income, Student, Credit_Rating, and Buy_Computer, formatted in Notepad with .arff file format.

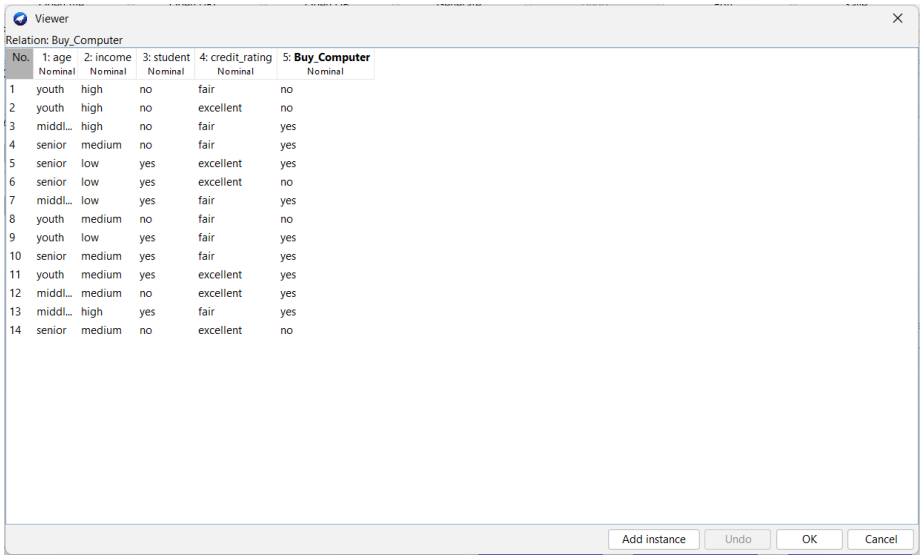
```
@relation Buy_Computer
@attribute age {youth, middle_age, senior}
@attribute income {low, medium, high}
@attribute student {yes, no}
@attribute credit_rating {excellent, fair}
@attribute Buy_Computer {yes, no}

@data
youth,high,no,fair,no
youth,high,no,excellent,no
middle_age,high,no,fair,yes
senior,medium,no,fair,yes
senior,low,yes,excellent,yes
senior,low,yes,excellent,no
middle_age,low,yes,fair,yes
youth,medium,no,fair,no
youth,low,yes,fair,yes
senior,medium,yes,fair,yes
youth,medium,yes,excellent,yes
middle_age,medium,no,excellent,yes
middle_age,high,yes,fair,yes
senior,medium,no,excellent,no
```

2. Select the dataset that we created earlier.

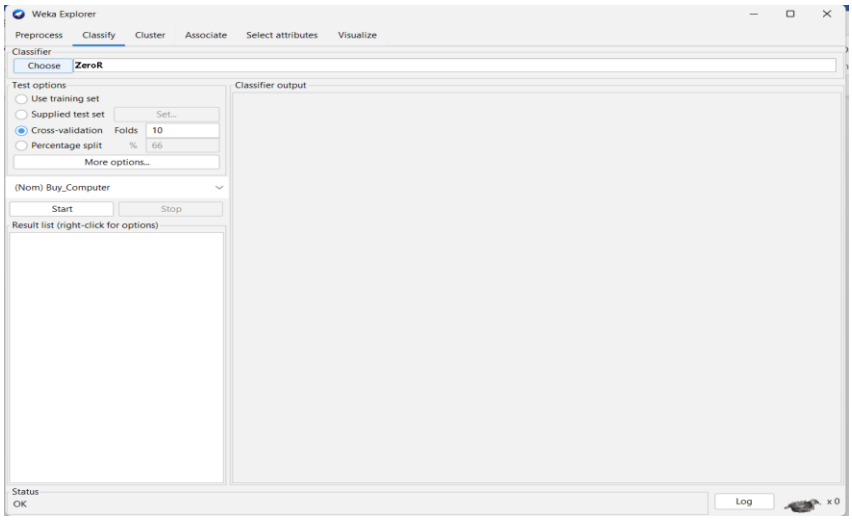




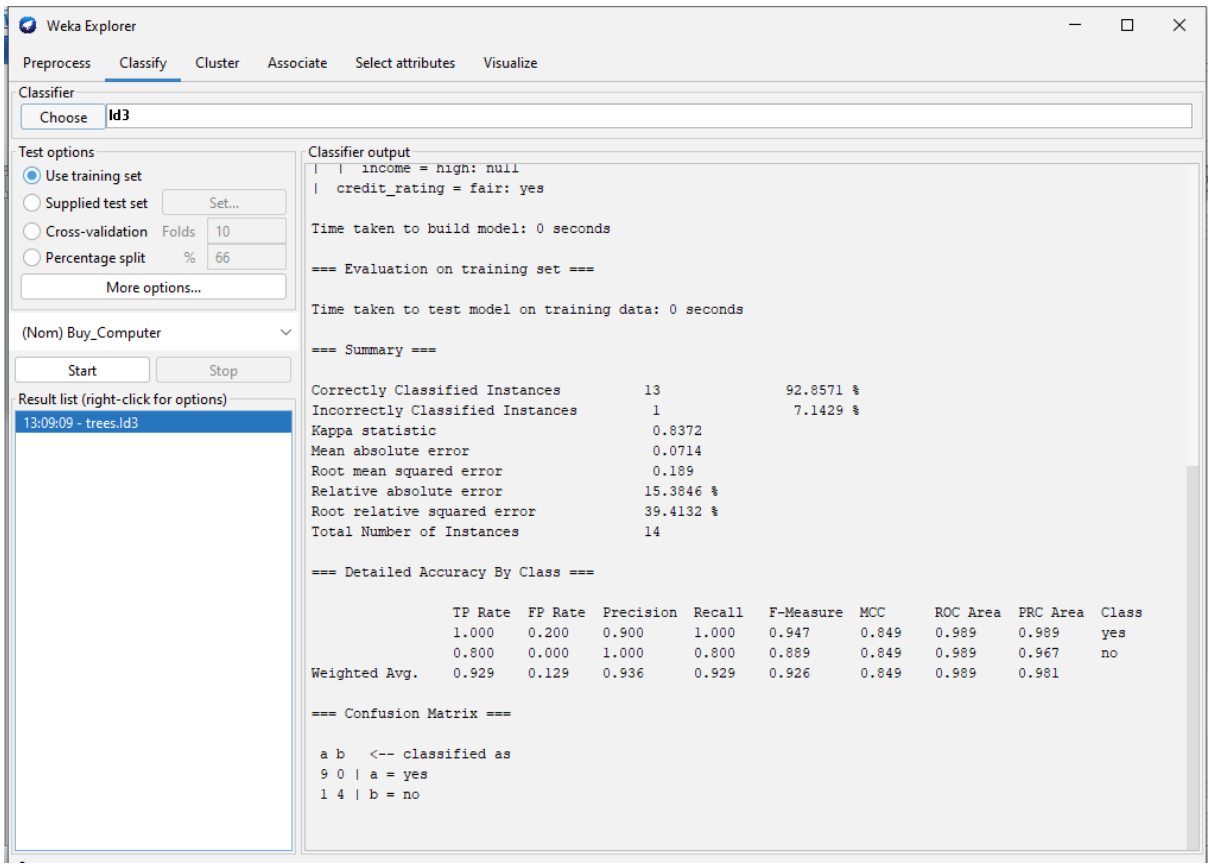
3. Please click "Edit" to view the Dataset in tabular format.



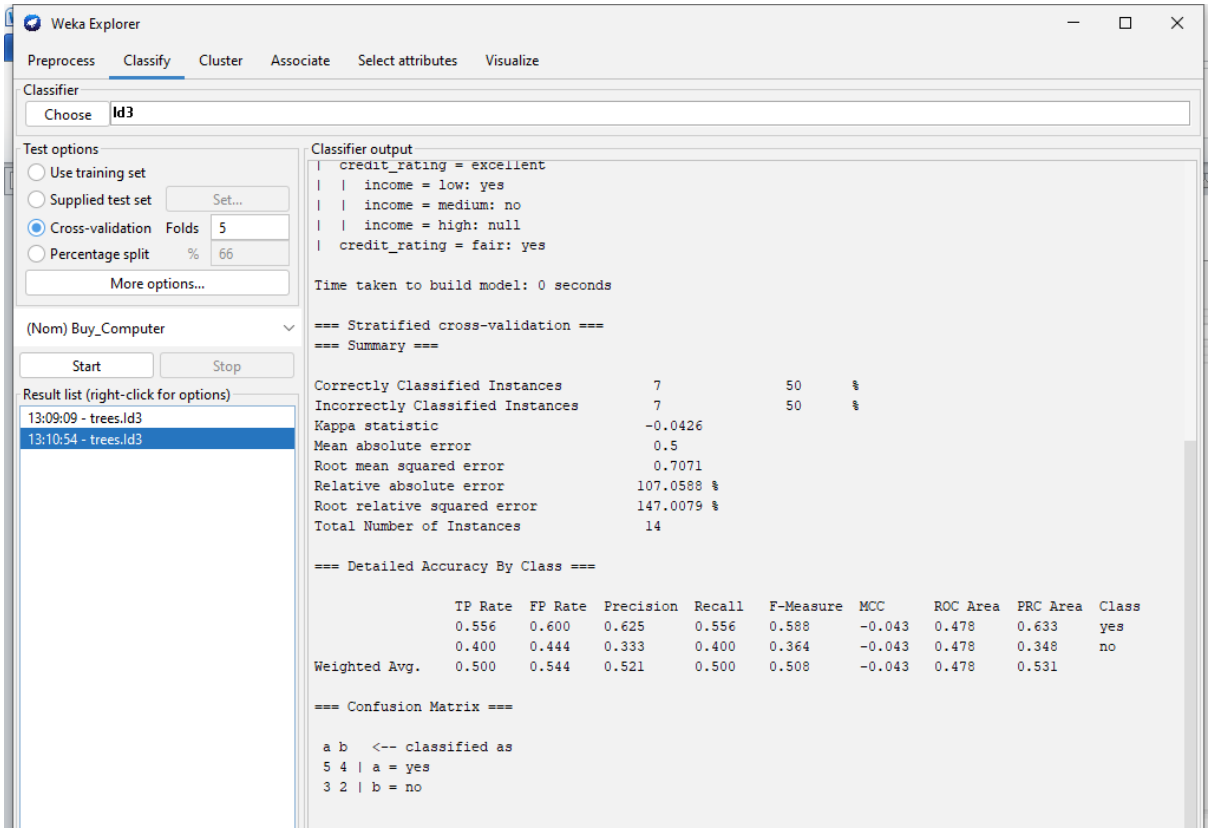
4. Please open the Classify window to utilize the ID3 Decision Tree algorithm for classifying the dataset.



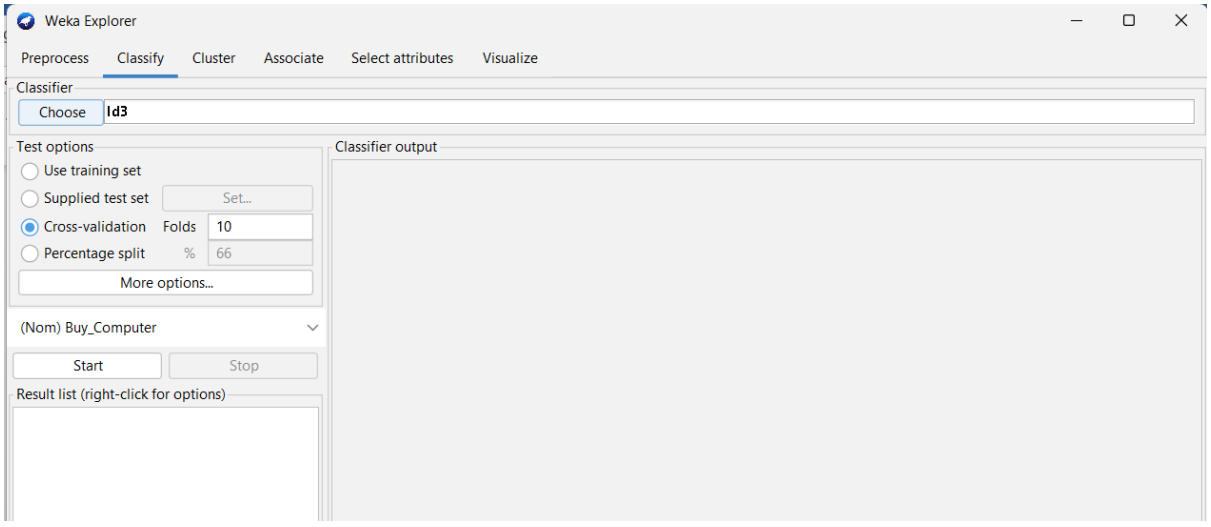
5. Click on "Choose" and select "trees > ID3" to choose the package for the ID3 Decision Tree algorithm.



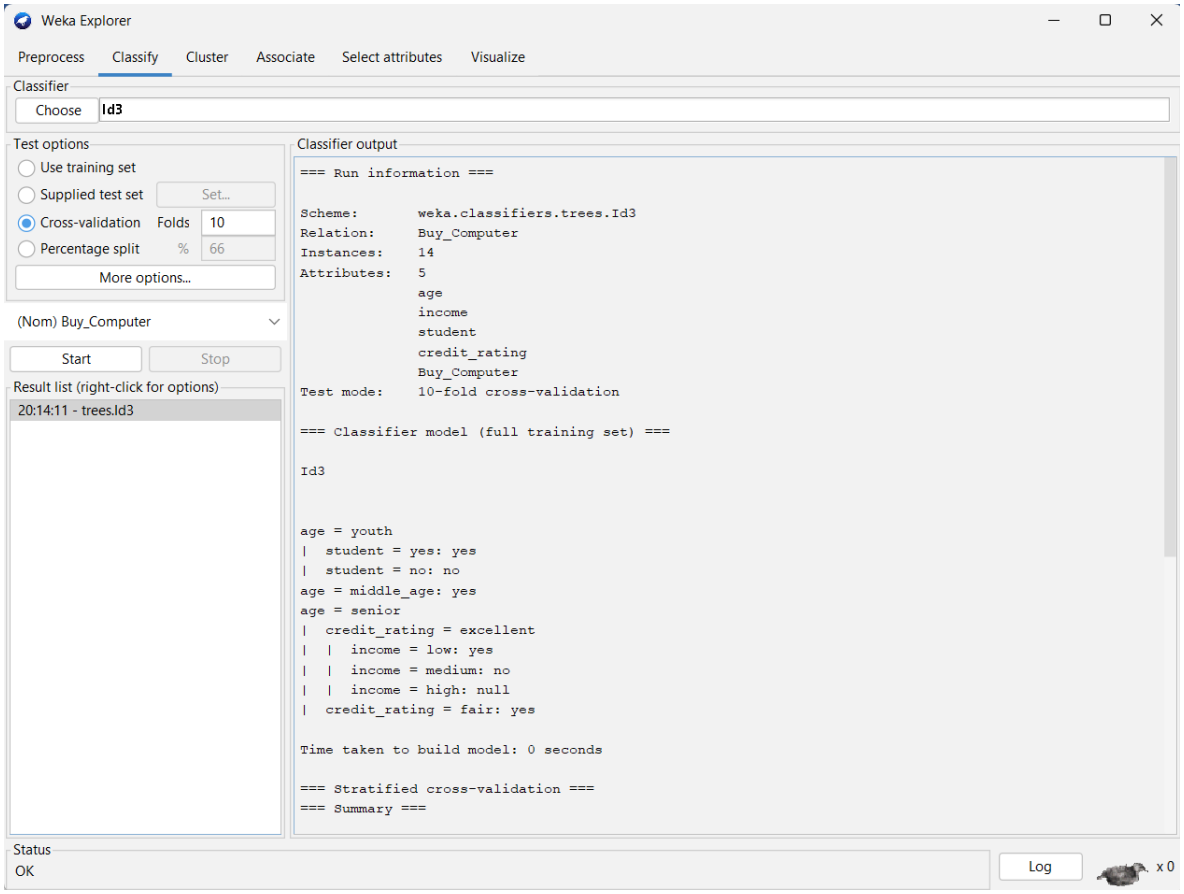
6. Switch the test options to Cross Validation mode (Folds : 5).

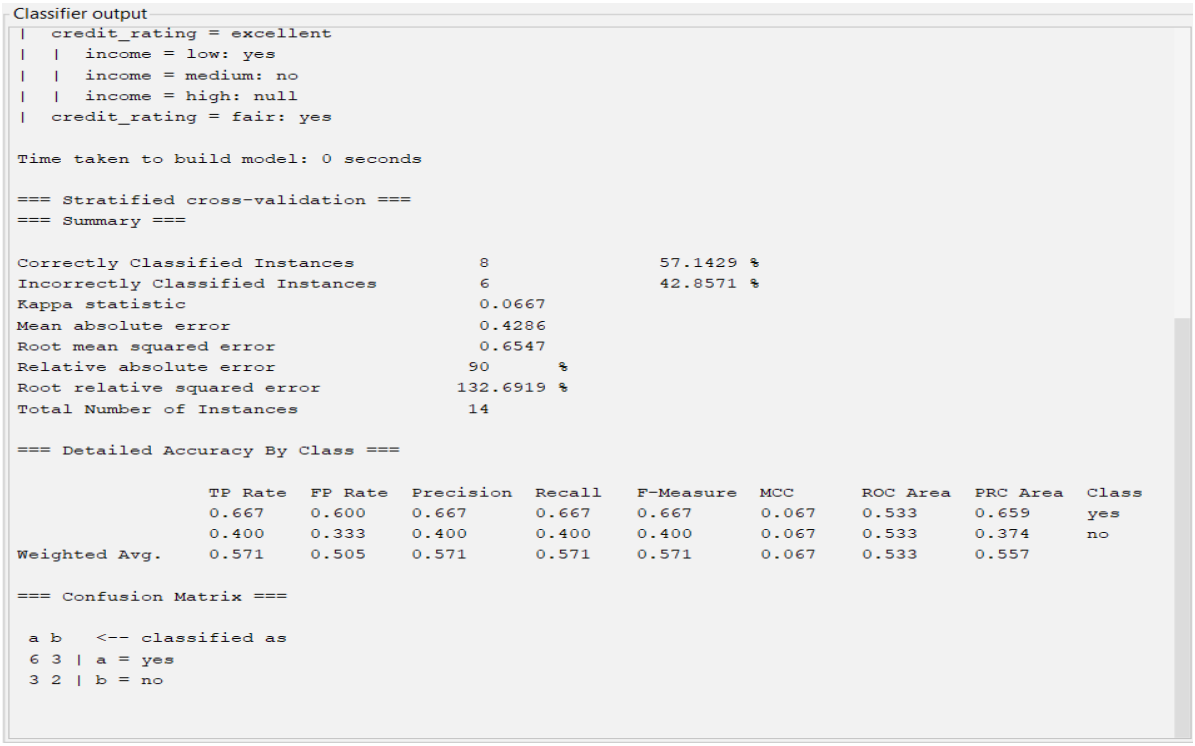


7. Switch the test options to Cross Validation mode (Folds : 10 Default).

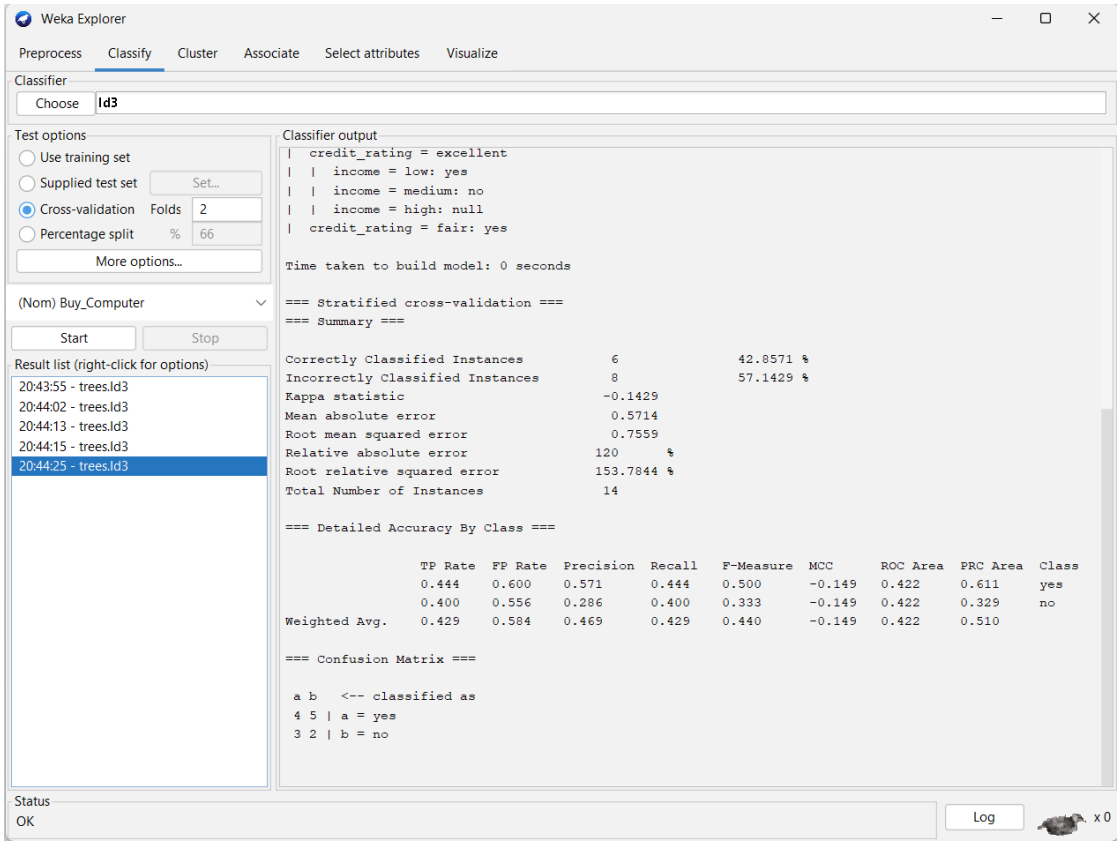


8. Initiate the classification process by clicking on the "Start" button.





9. Modify the fold parameter from 10 to any random number between 1 and 14, considering that there are 14 instances in the dataset.



10. To obtain varying outputs, alter the fold parameter multiple times.

Experiment-7

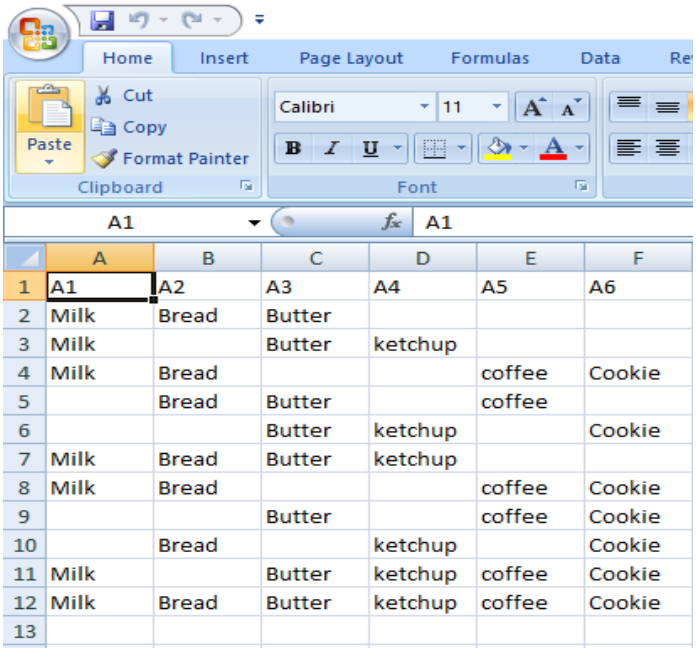
Aim: Generate frequent item set using Apriori Algorithm and also generate association rules for any market basket data

Description:

Apriori is the first association rule mining algorithm that use of support-based pruning to systematically control the exponential growth of candidate item sets.
And the apriori principle states that"If an item set is frequent, then all of its subsets must also be frequent".

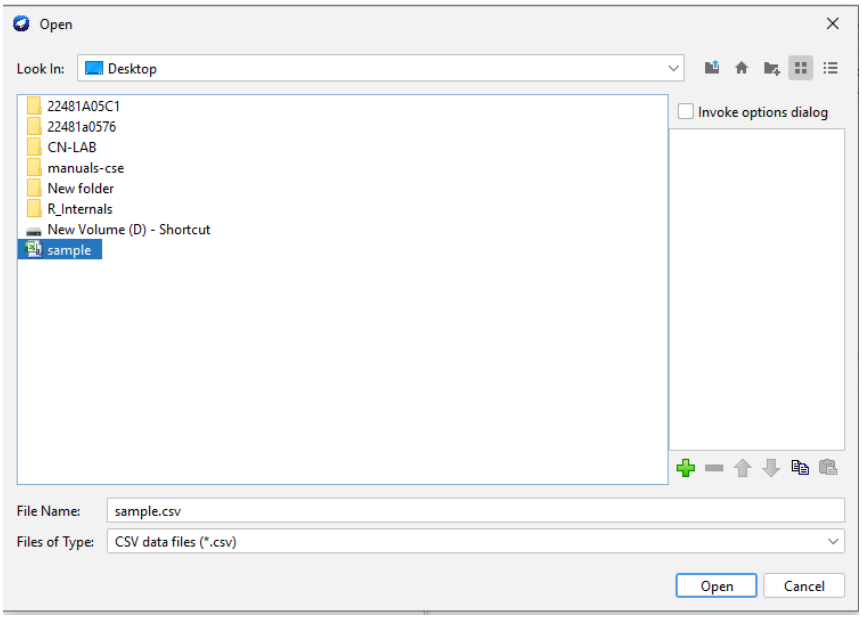
Process:

1. First create the market basket data(transaction data set) and save it in the CSV form

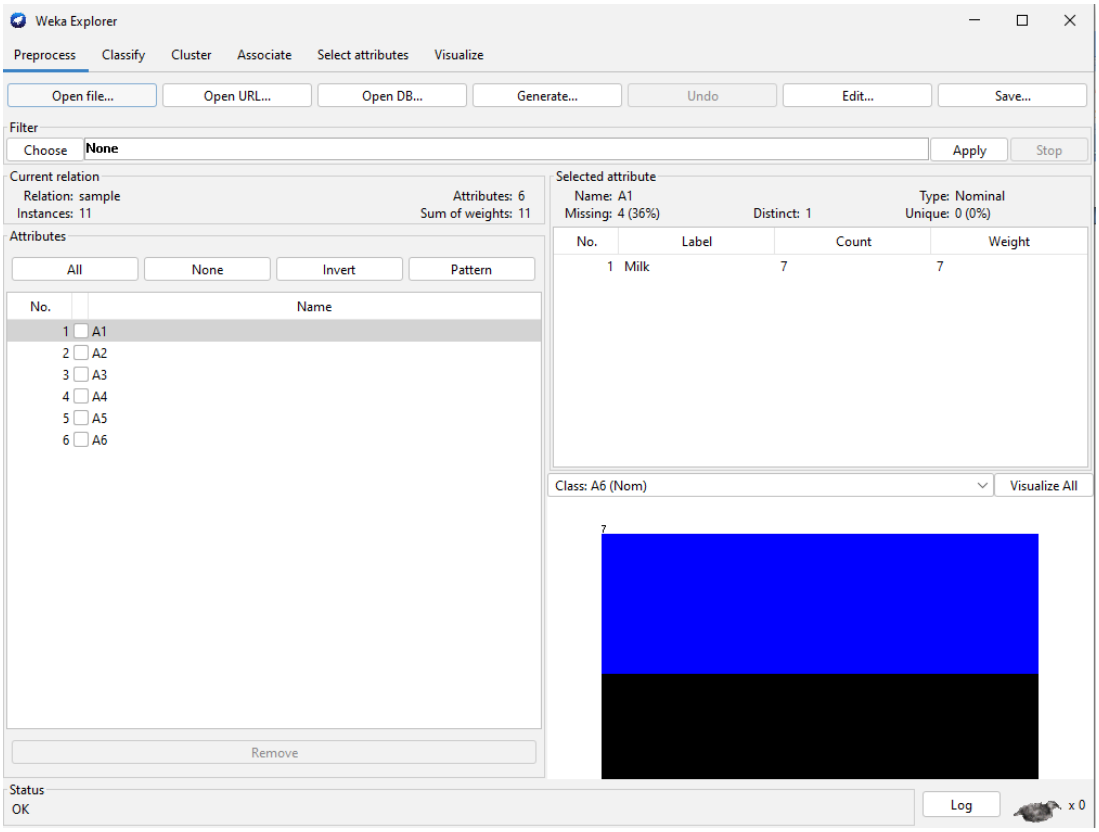


	A	B	C	D	E	F
1	A1	A2	A3	A4	A5	A6
2	Milk	Bread	Butter			
3	Milk		Butter	ketchup		
4	Milk	Bread			coffee	Cookie
5		Bread	Butter		coffee	
6			Butter	ketchup		Cookie
7	Milk	Bread	Butter	ketchup		
8	Milk	Bread			coffee	Cookie
9			Butter		coffee	Cookie
10		Bread		ketchup		Cookie
11	Milk		Butter	ketchup	coffee	Cookie
12	Milk	Bread	Butter	ketchup	coffee	Cookie
13						

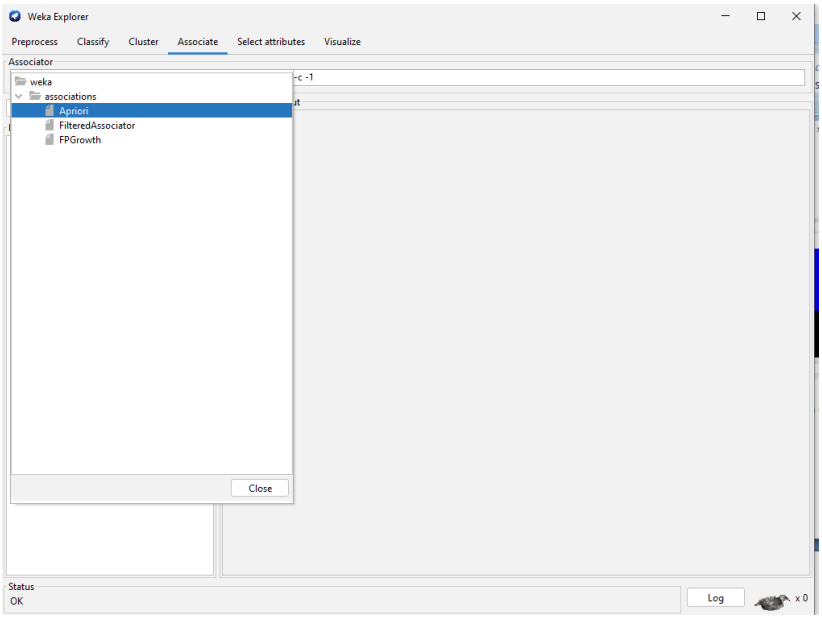
2. Open weka tool>explore>and load the dataset which has be created



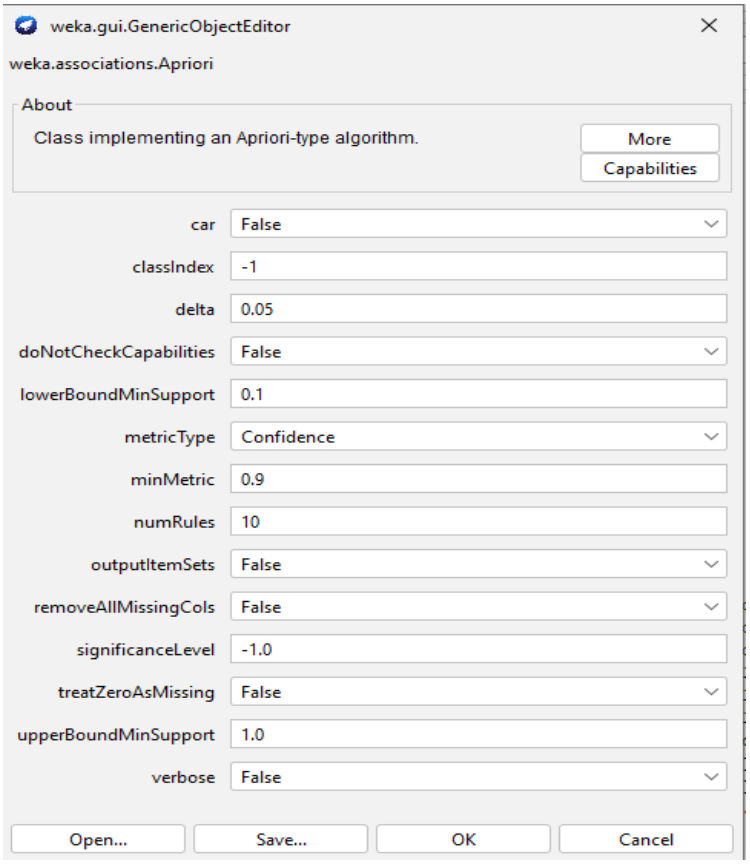
3. Once the dataset is loaded you will observe the window as shown below

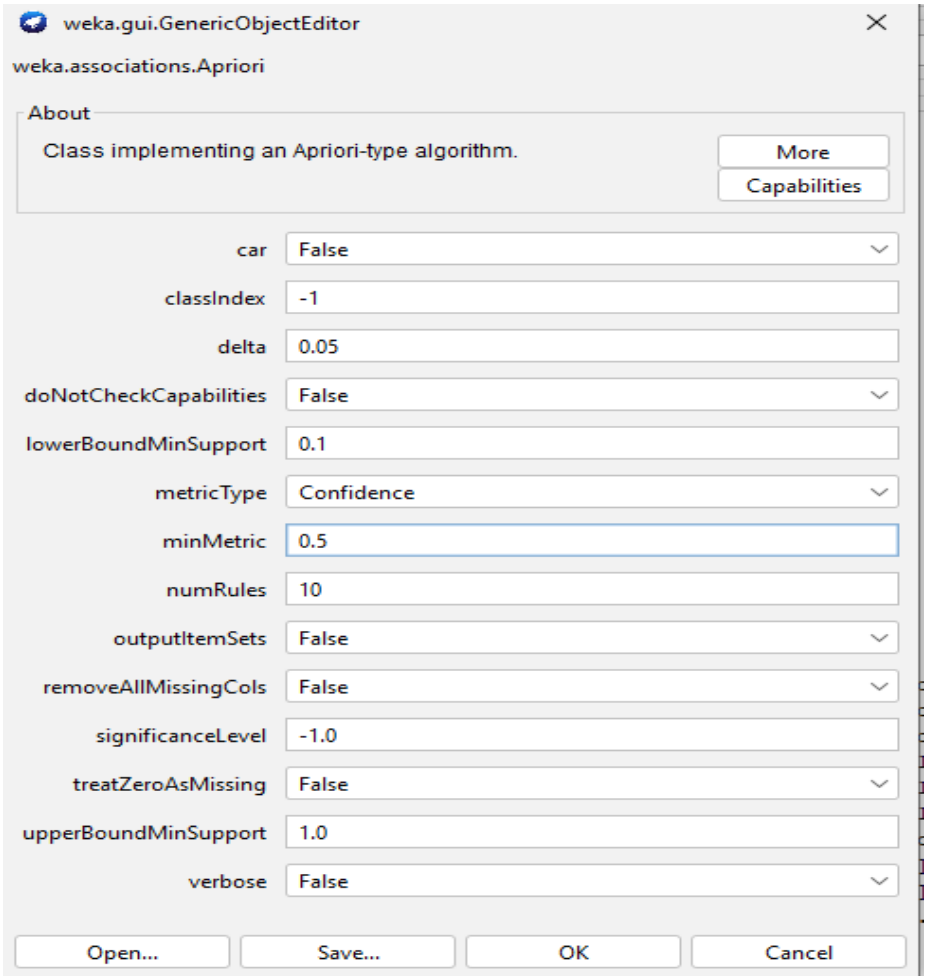


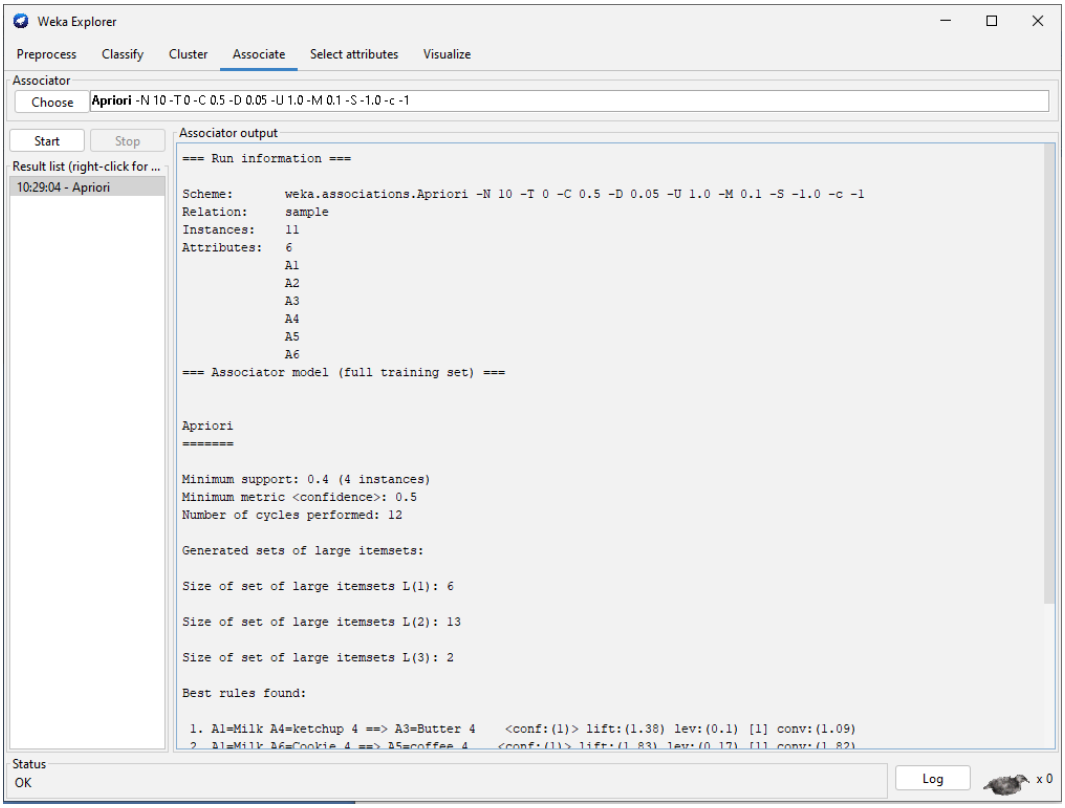
4. After this as we have to apply the association rules go to associate and click on choose and select the Apriori algorithm

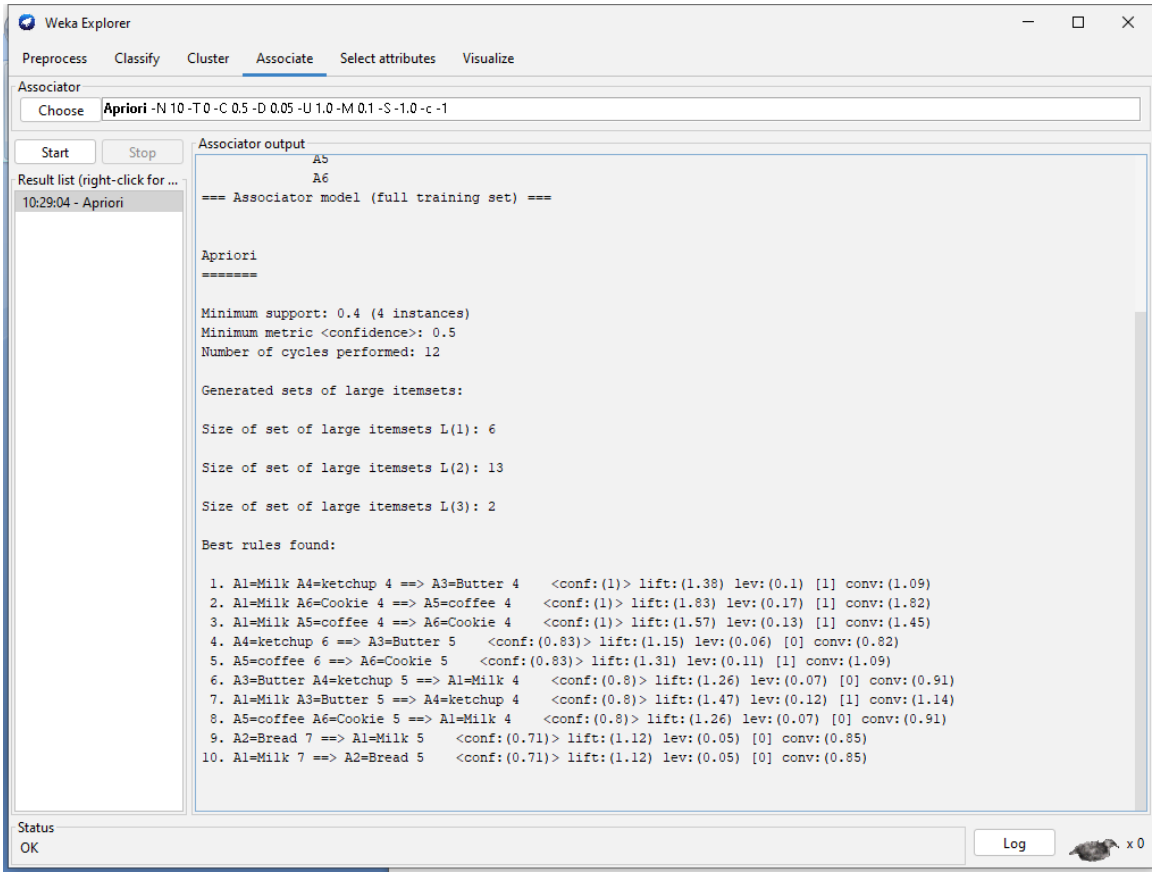


5. Once you click on the Apriori you get a window where you can adjust the features such as support and confidence. Adjust these features and now apply to see the output.




weka.gui.GenericObjectEditor
weka.associations.Apriori
About
Class implementing an Apriori-type algorithm.
More
Capabilities
car
False
classIndex
-1
delta
0.05
doNotCheckCapabilities
False
lowerBoundMinSupport
0.1
metricType
Confidence
minMetric
0.5
numRules
10
outputItemSets
False
removeAllMissingCols
False
significanceLevel
-1.0
treatZeroAsMissing
False
upperBoundMinSupport
1.0
verbose
False
Open...
Save...
OK
Cancel


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Associator
Choose
Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Start
Stop
Result list (right-click for ...
10:29:04 - Apriori
Associator output
=== Run information ===
Scheme: weka.associations.Apriori -N 10 -T 0 -C 0.5 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation: sample
Instances: 11
Attributes: 6
A1
A2
A3
A4
A5
A6
=== Associator model (full training set) ===
Apriori
=======
Minimum support: 0.4 (4 instances)
Minimum metric <confidence>: 0.5
Number of cycles performed: 12
Generated sets of large itemsets:
Size of set of large itemsets L(1): 6
Size of set of large itemsets L(2): 13
Size of set of large itemsets L(3): 2
Best rules found:
1. A1=Milk A4=ketchup 4 ==> A3=Butter 4 <conf:(1)> lift:(1.38) lev:(0.1) [1] conv:(1.09)
Status
OK
Log
x 0



6. The output window results the below sections
    I. Run information
    II. Associator Model(aproiri)
    III. Best rules found

Experiment -8

Aim: Generate frequent item set using FP growth Algorithm

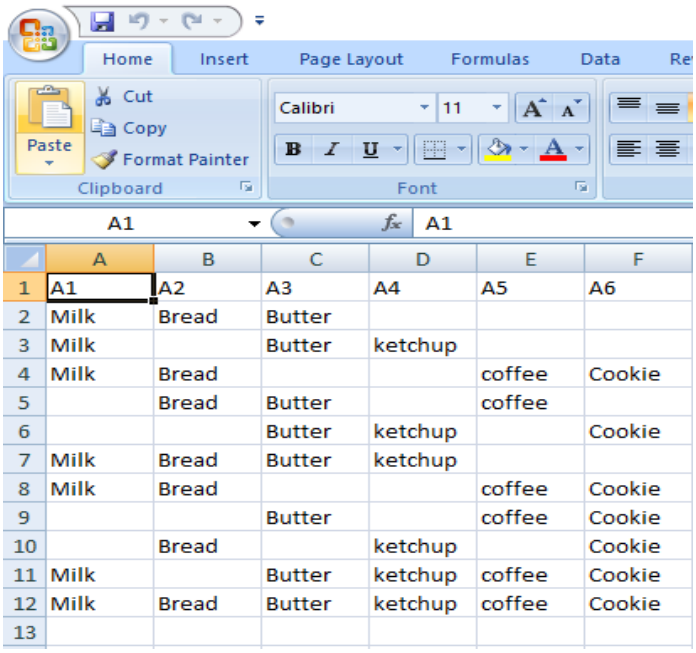
Description:

FP growth encodes the data set using a **compact data structure** called an FP-tree and extracts **frequent itemsets** directly from this structure.
And the representation of the FP tree is as follows:

- An FP-tree is a compressed representation of the input data.
- It is constructed by reading the data set one transaction at a time and mapping each transaction onto a path in the FP-tree.
- As different transactions have several items in common, **their paths may overlap**

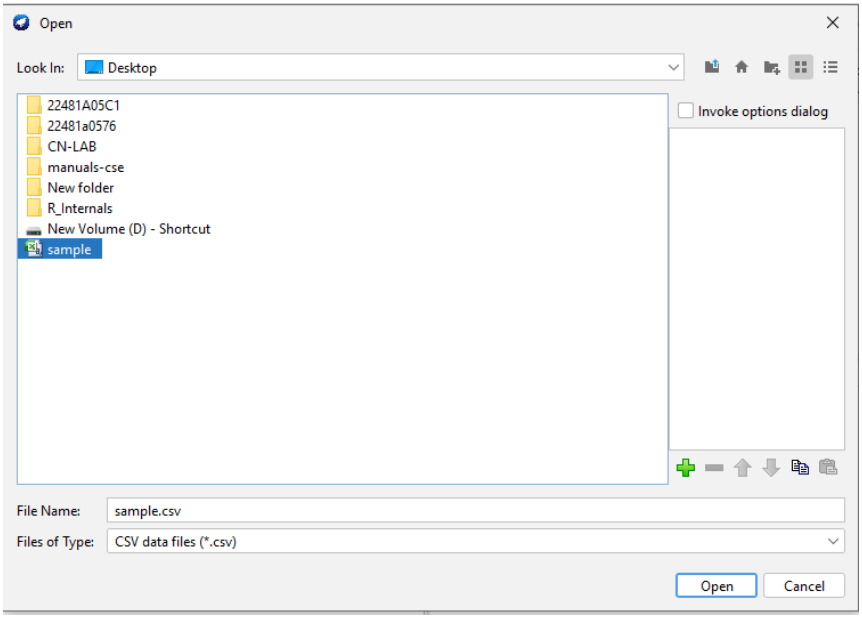
Process:

7. First create the market basket data(transaction data set) and save it in the CSV form

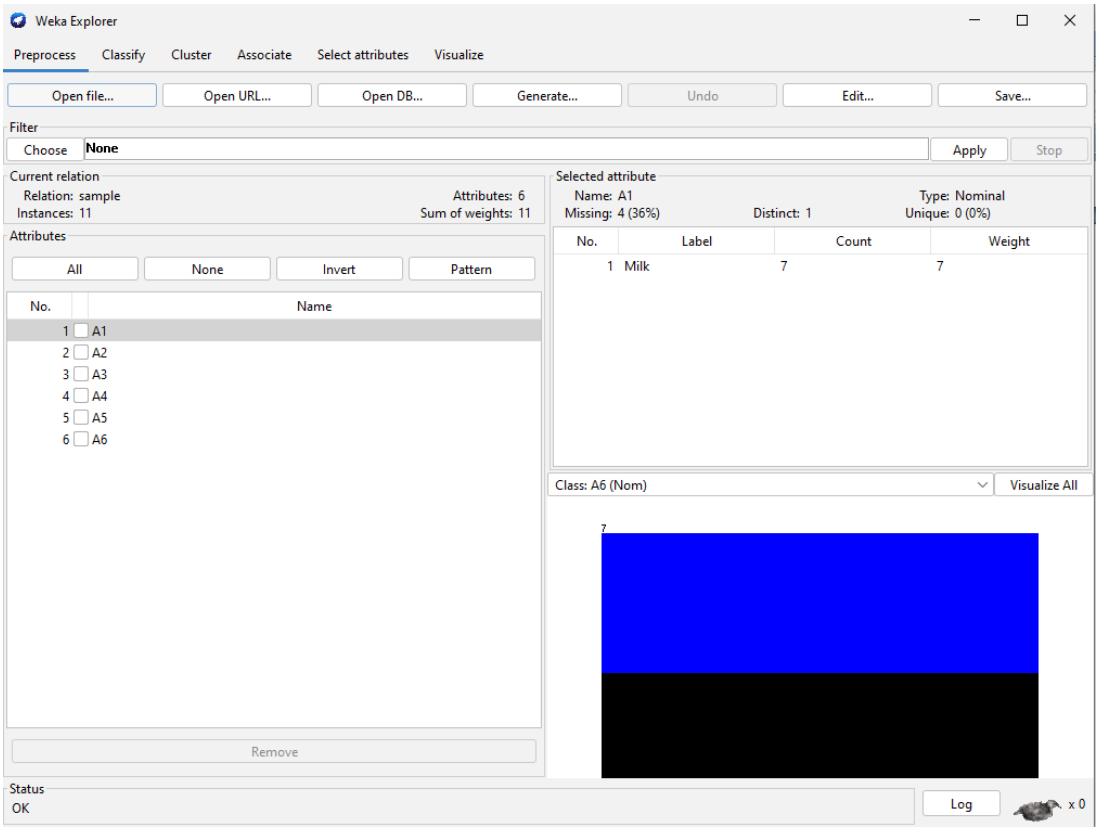


	A	B	C	D	E	F
1	A1	A2	A3	A4	A5	A6
2	Milk	Bread	Butter			
3	Milk		Butter	ketchup		
4	Milk	Bread			coffee	Cookie
5		Bread	Butter		coffee	
6			Butter	ketchup		Cookie
7	Milk	Bread	Butter	ketchup		
8	Milk	Bread			coffee	Cookie
9			Butter		coffee	Cookie
10		Bread		ketchup		Cookie
11	Milk		Butter	ketchup	coffee	Cookie
12	Milk	Bread	Butter	ketchup	coffee	Cookie
13						

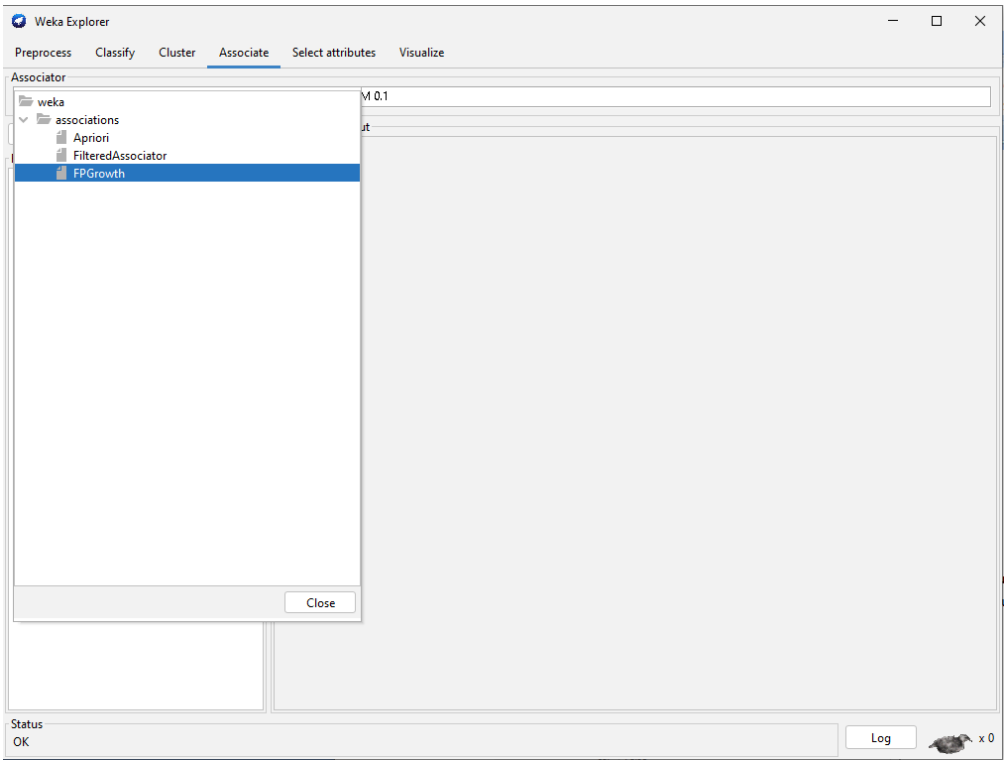
8. Open weka tool>explore>and load the dataset which has be created



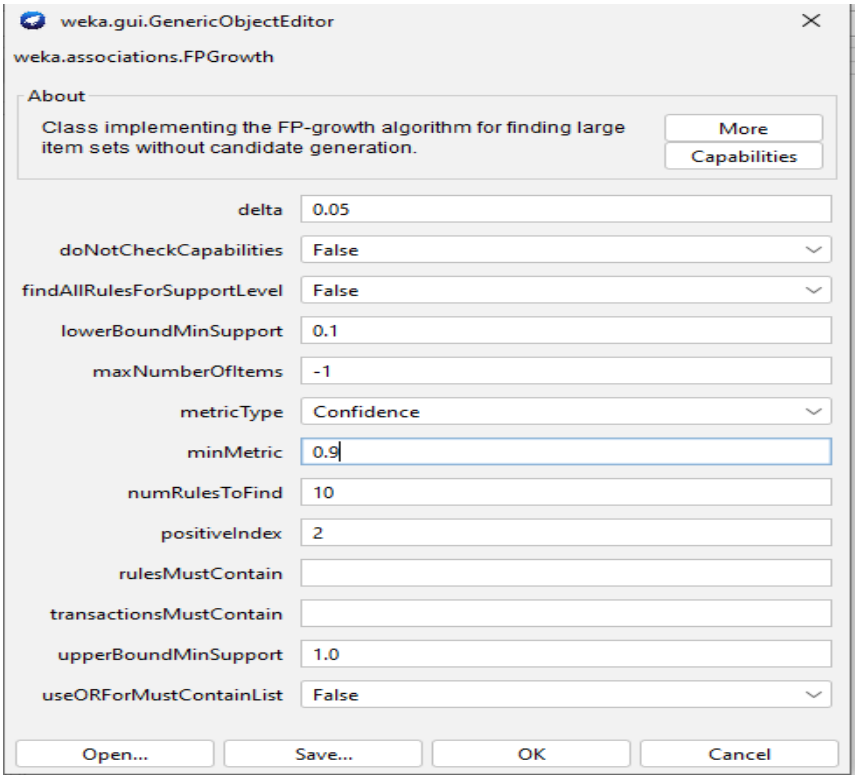
9. Once the dataset is loaded you will observe the window as shown below

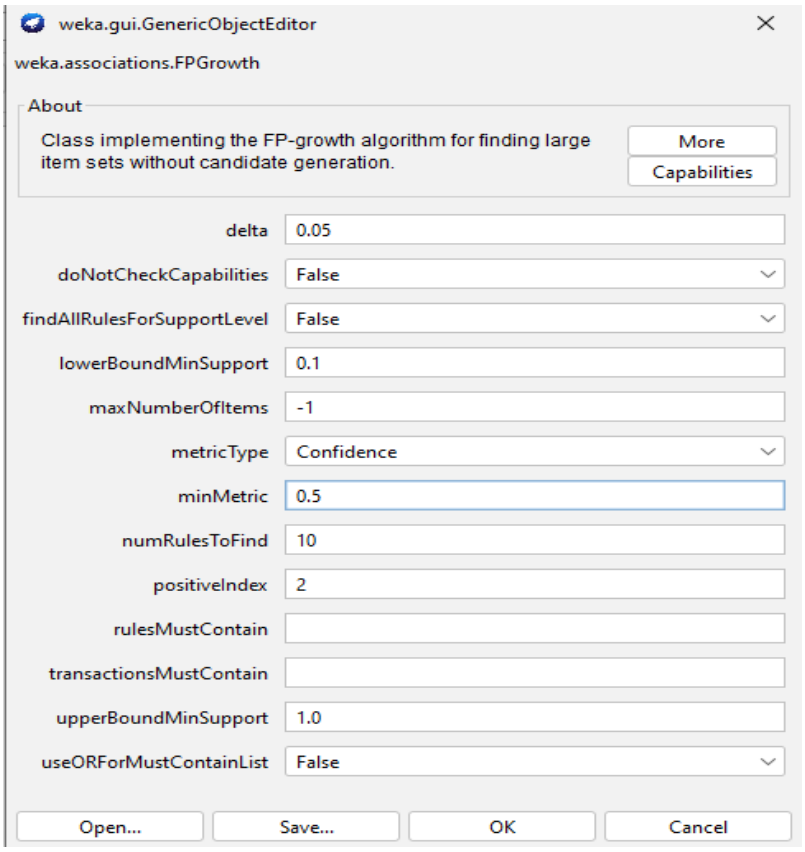


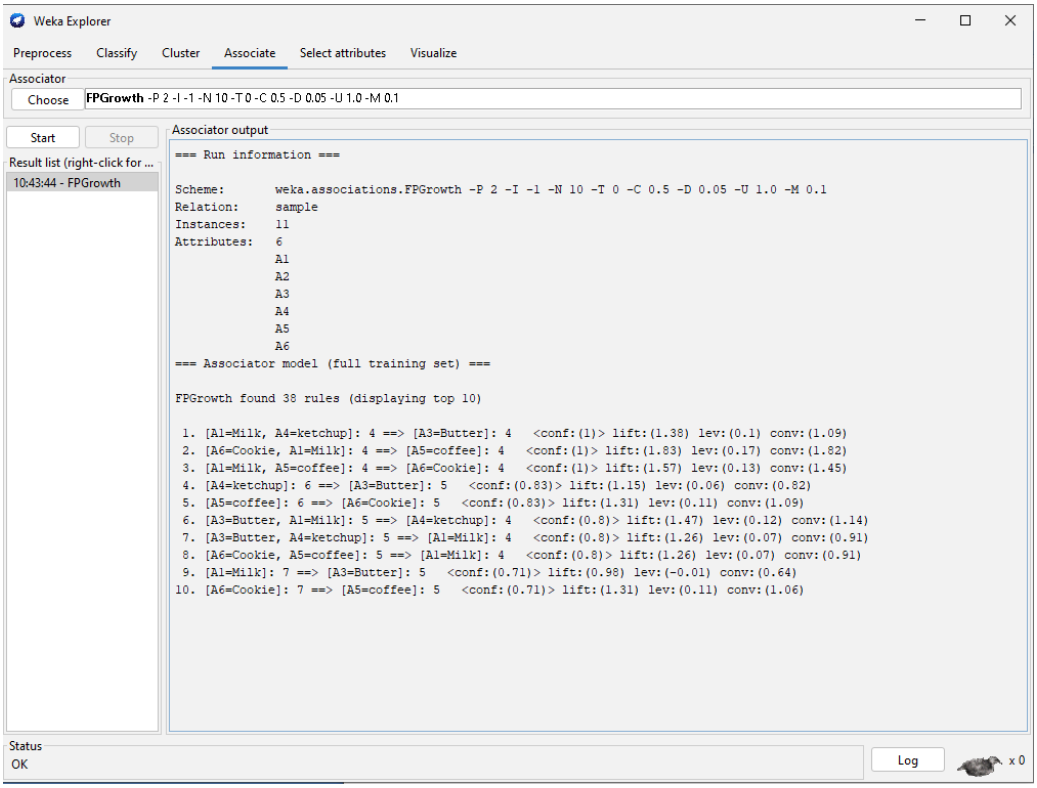
10. After this as we have to apply the association rules go to associate and click on choose and select the FP growth algorithm



11. Once you click on the FP growth you get a window where you can adjust the features such as support and confidence. Adjust these features and now apply to see the output.


weka.gui.GenericObjectEditor
weka.associations.FPGrowth
About
Class implementing the FP-growth algorithm for finding large item sets without candidate generation.
More
Capabilities
delta 0.05
doNotCheckCapabilities False
findAllRulesForSupportLevel False
lowerBoundMinSupport 0.1
maxNumberOfItems -1
metricType Confidence
minMetric 0.9
numRulesToFind 10
positiveIndex 2
rulesMustContain
transactionsMustContain
upperBoundMinSupport 1.0
useORForMustContainList False
Open...
Save...
OK
Cancel


weka.gui.GenericObjectEditor
weka.associations.FPGrowth
About
Class implementing the FP-growth algorithm for finding large item sets without candidate generation.
More
Capabilities
delta 0.05
doNotCheckCapabilities False
findAllRulesForSupportLevel False
lowerBoundMinSupport 0.1
maxNumberOfItems -1
metricType Confidence
minMetric 0.5
numRulesToFind 10
positiveIndex 2
rulesMustContain
transactionsMustContain
upperBoundMinSupport 1.0
useORForMustContainList False
Open...
Save...
OK
Cancel



12. The output window results the below sections
    - IV. Run information
    - V. Associator Model(FP growth)
    - VI. Best rules found

Experiment -9

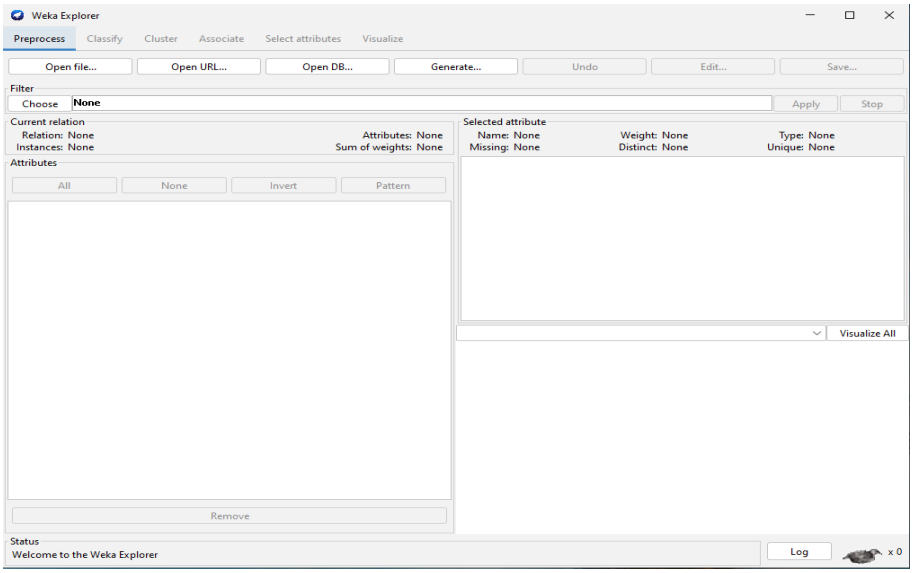
Aim: Experiment to compare the performance of various data mining algorithm on any data set

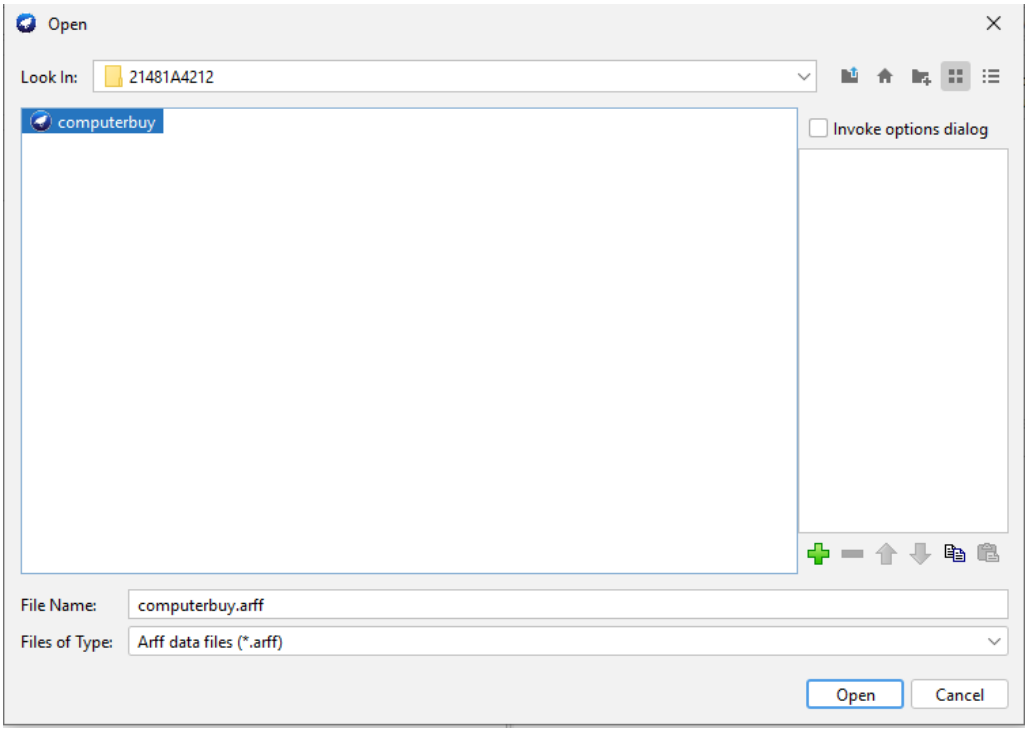
Description:

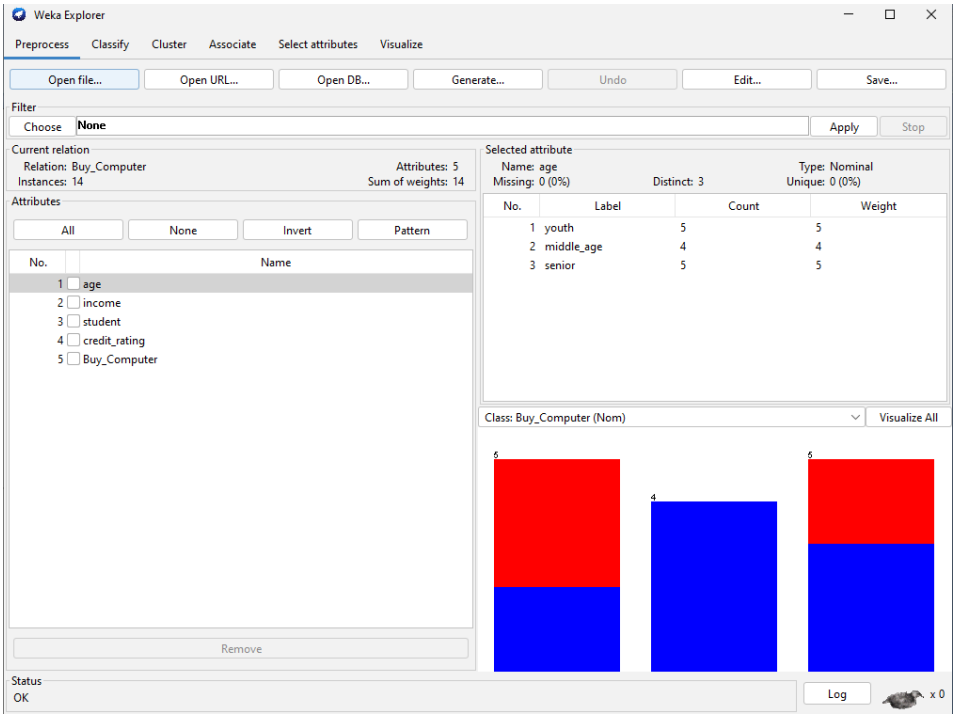
Data Mining Algorithms are a particular category of algorithms useful for analyzing data and developing data models to identify meaningful patterns. These are part of machine learning algorithms. These algorithms are implemented through various programming like R language, Python, and data mining tools to derive the optimized data models. Some of the data mining algorithms are C4.5 for decision trees, K-means for cluster data analysis, Naive Bayes Algorithm and many more and these algorithms are based upon statistical and mathematical formulas which applied to the data set.

Process:

1. Open weka tool >explore>choose file and load the .arrf file


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: None
Instances: None
Attributes: None
Sum of weights: None
Selected attribute
Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None
Attributes
All
None
Invert
Pattern
Visualize All
Remove
Status
Welcome to the Weka Explorer
Log
x 0


Open
Look In:
21481A4212
computerbuy
Invoke options dialog
File Name:
computerbuy.arff
Files of Type:
Arff data files (*.arff)
Open
Cancel

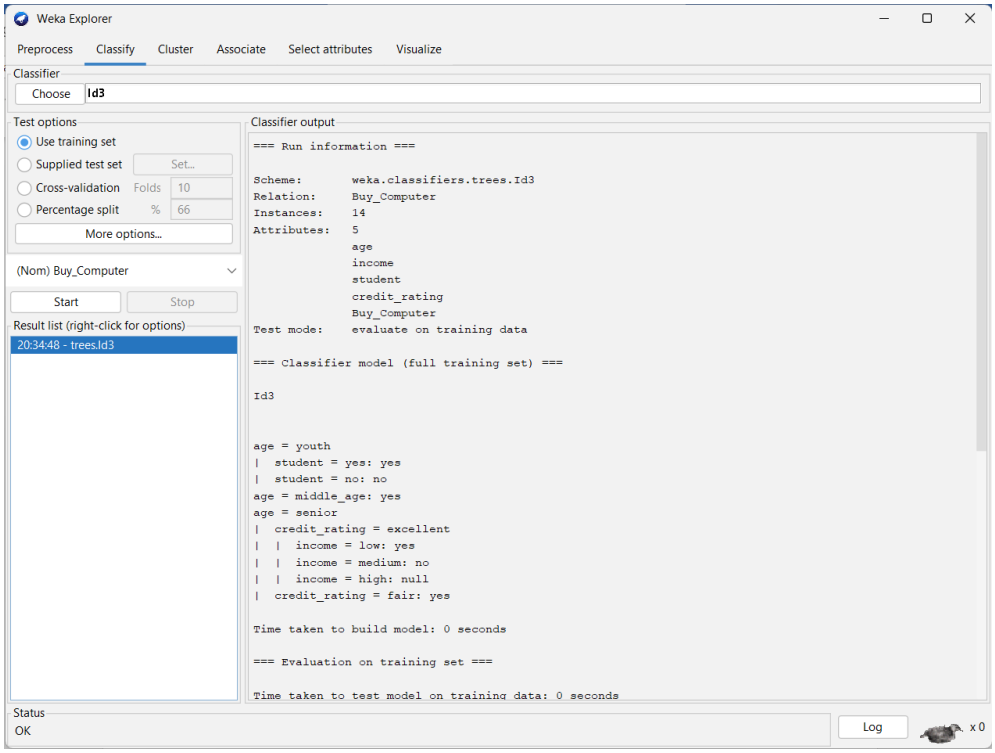


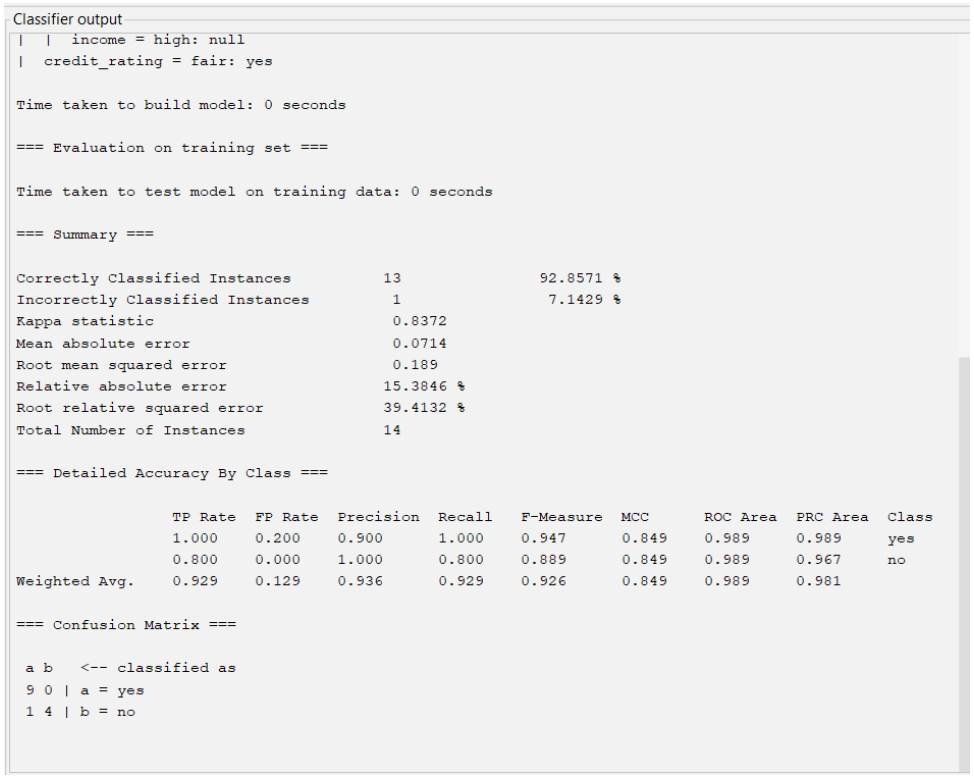
2. One of the outputs of the classifications algorithm is confusion matrix and the general structure looks like as shown below

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

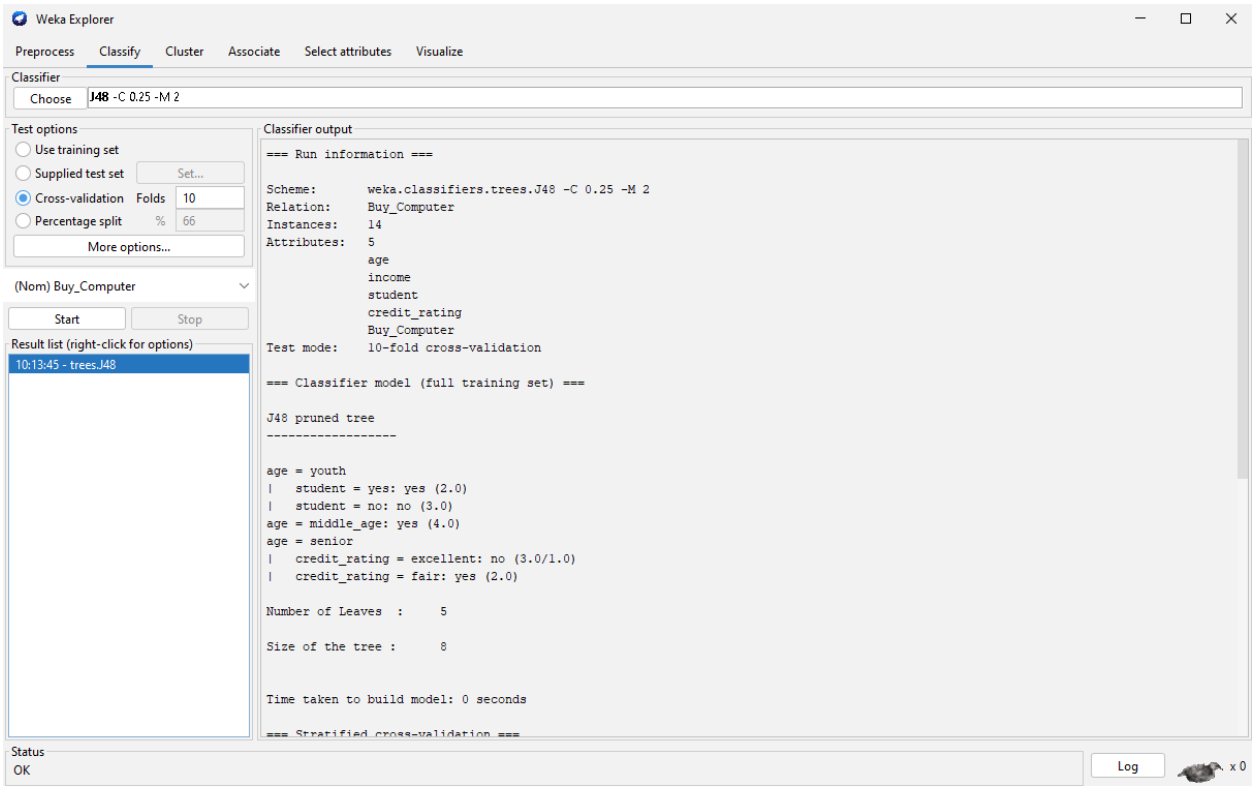
3. Now perform the below classification algorithms on the data set
   I. ID3 classification
   II. J48 classification
   III. Navies Bayes classification

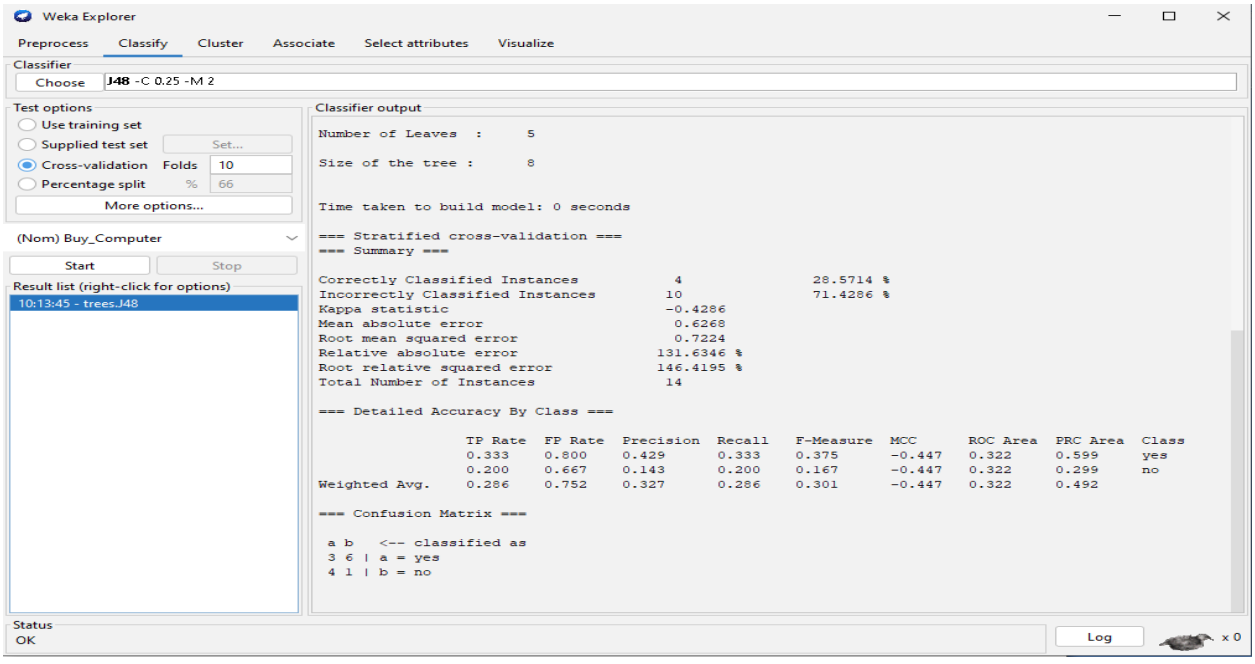
4. The output window of the ID3 classification is shown below



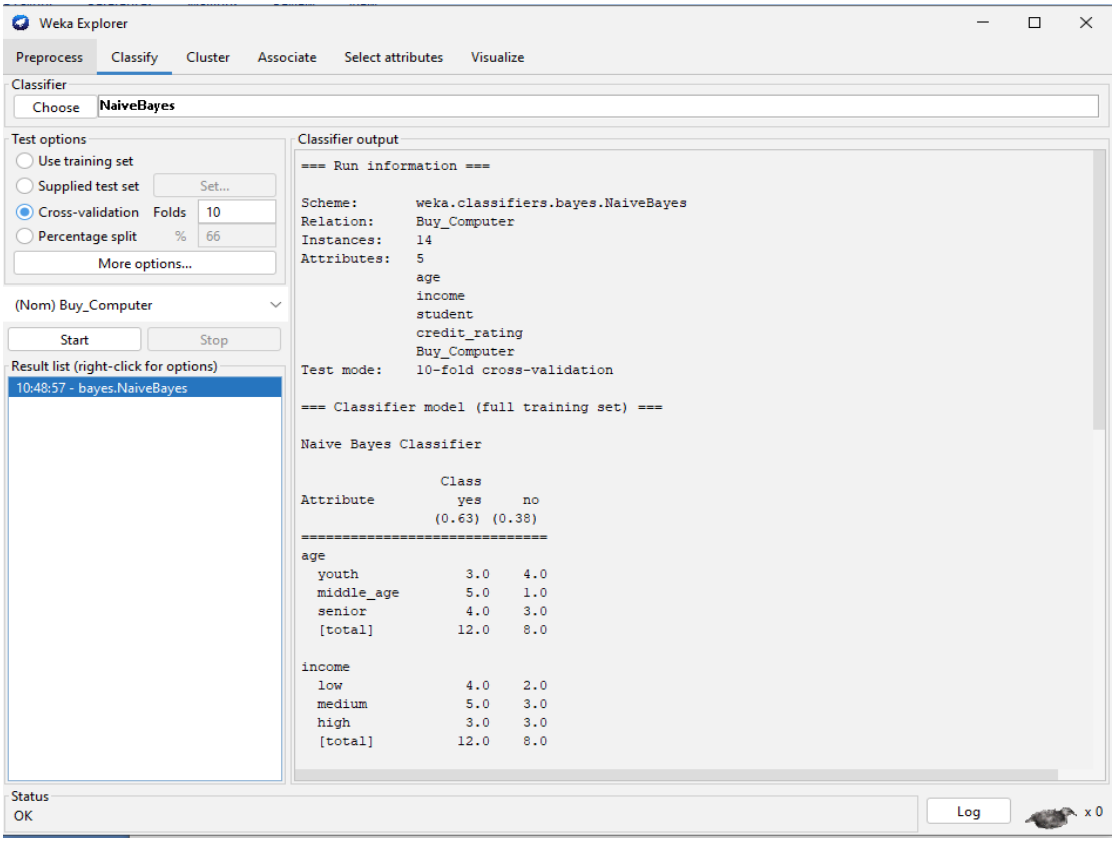


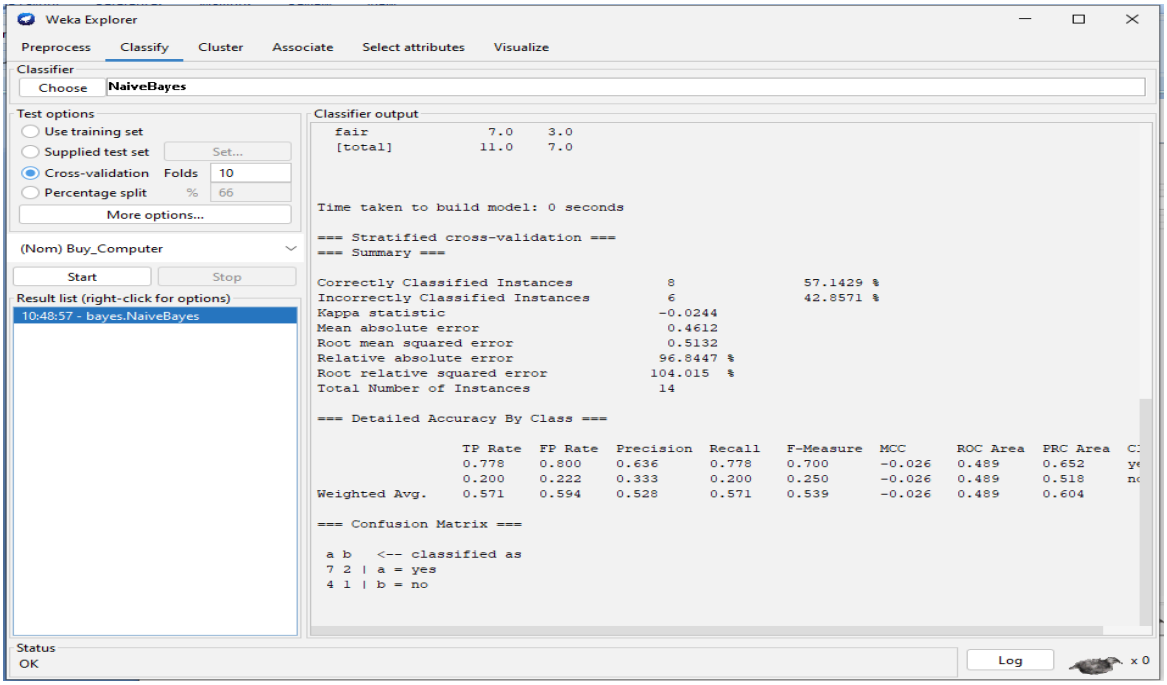
5. The output window of the J48 classification is shown below





6. The output window of the J48 classification is shown below





7. Now compare the accuracy of the three algorithms and the accuracy of ID3 has maximum compared to all classification algorithms that we had performed.

Experiment – 10

Aim:

Apply K-Means clustering algorithm on any data set by passing different k values.

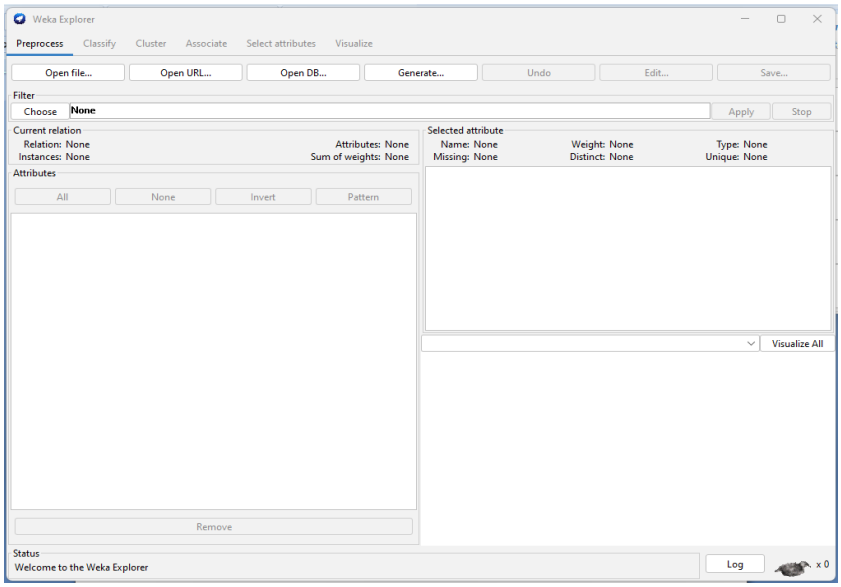
Description:

K means clustering, assigns data points to one of the K clusters depending on their distance from the center of the clusters. It starts by randomly assigning the clusters centroid in the space. Then each data point assign to one of the cluster based on its distance from centroid of the cluster. After assigning each point to one of the cluster, new cluster centroids are assigned. This process runs iteratively until it finds good cluster. In the analysis we assume that number of cluster is given in advanced and we have to put points in one of the group.

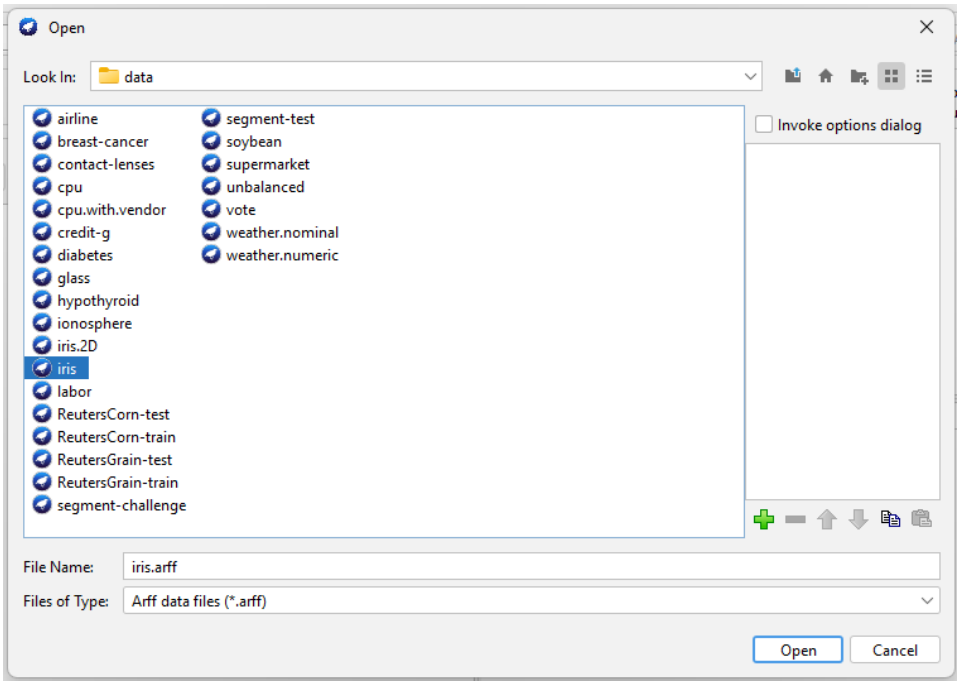
Process:

1. Open Weka tool

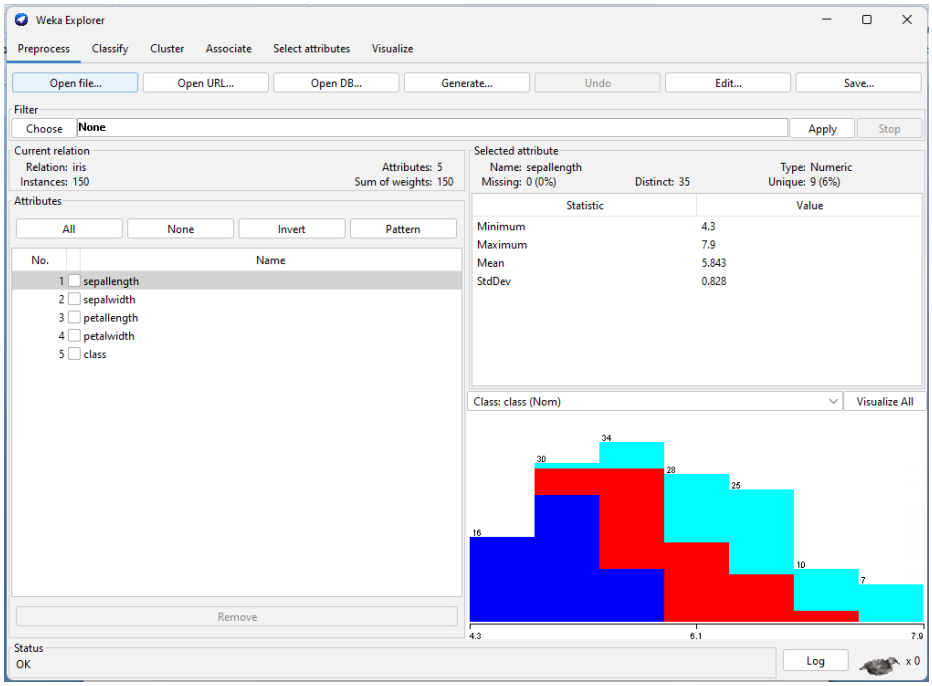
2. Goto Explorer



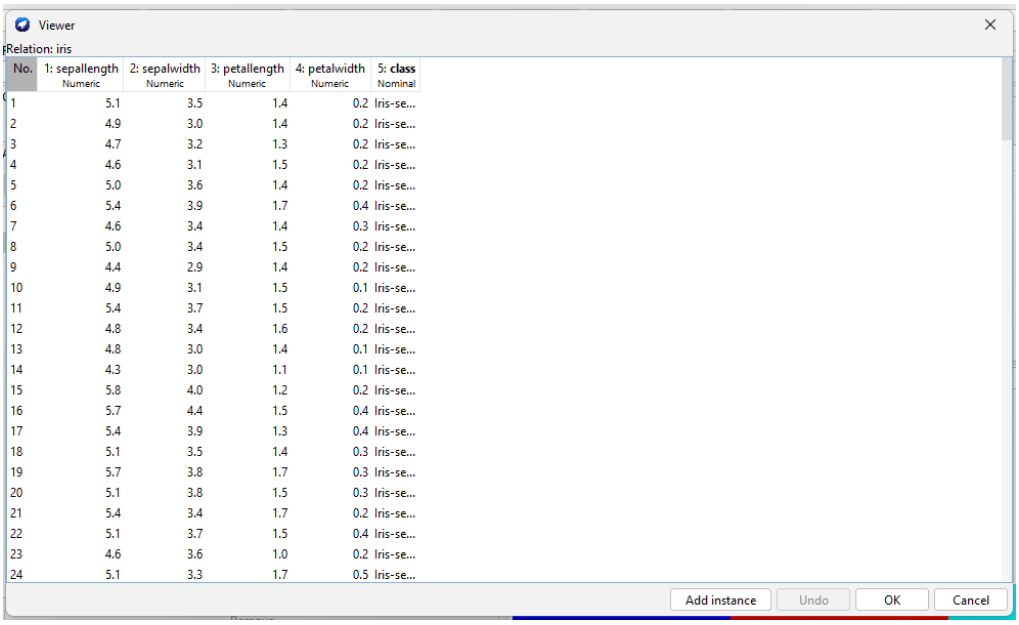
3. Click on “Open file”
4. Select the dataset which we have downloaded earlier (iris dataset)



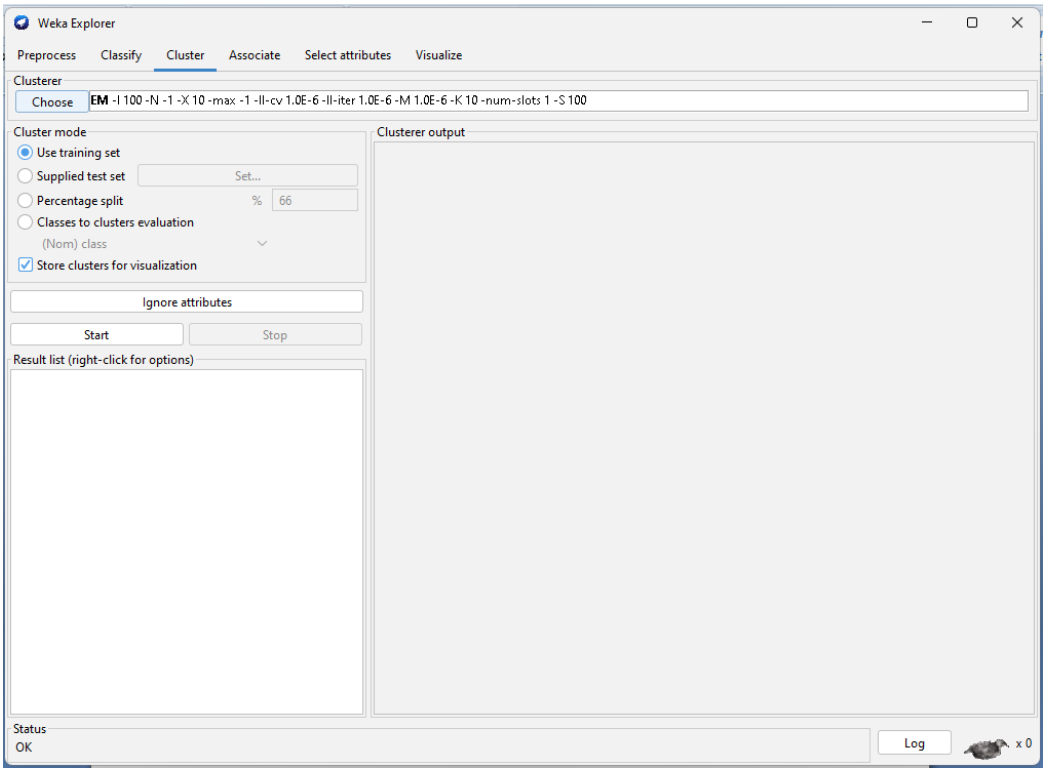
5. Click on “Open”



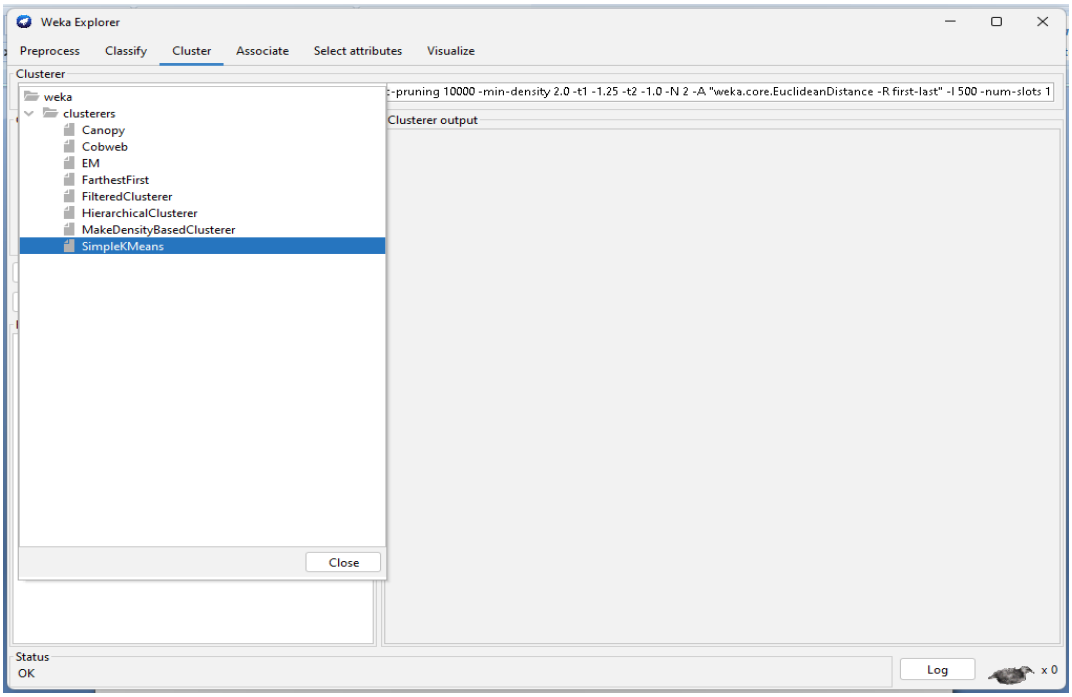
6. Click on “Edit” to view the dataset



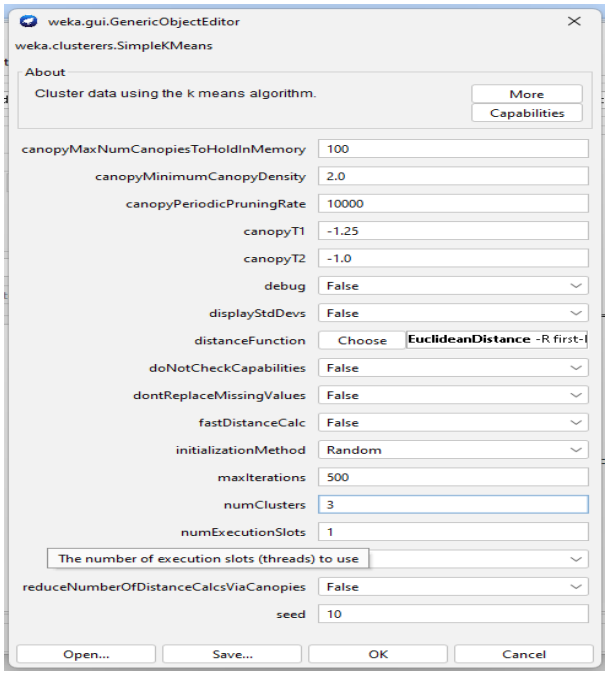
7. Now, click on “Cluster”



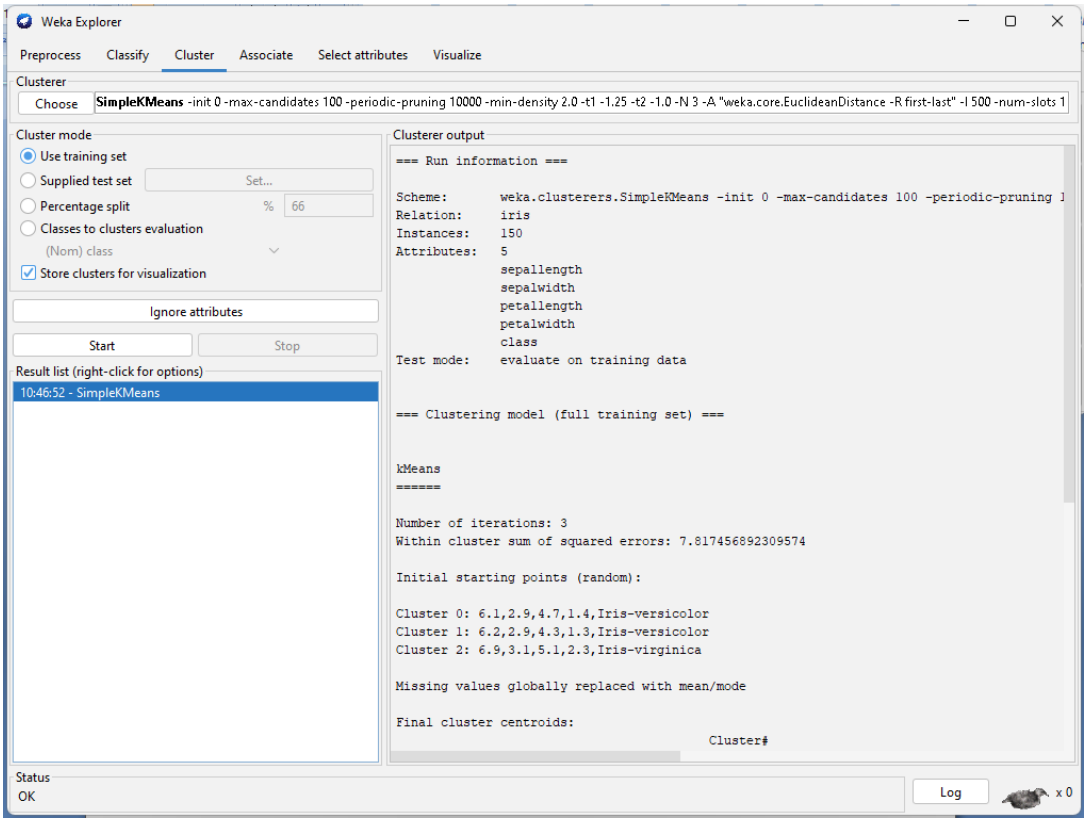
8. Goto “Choose” and select “SimpleKMeans”

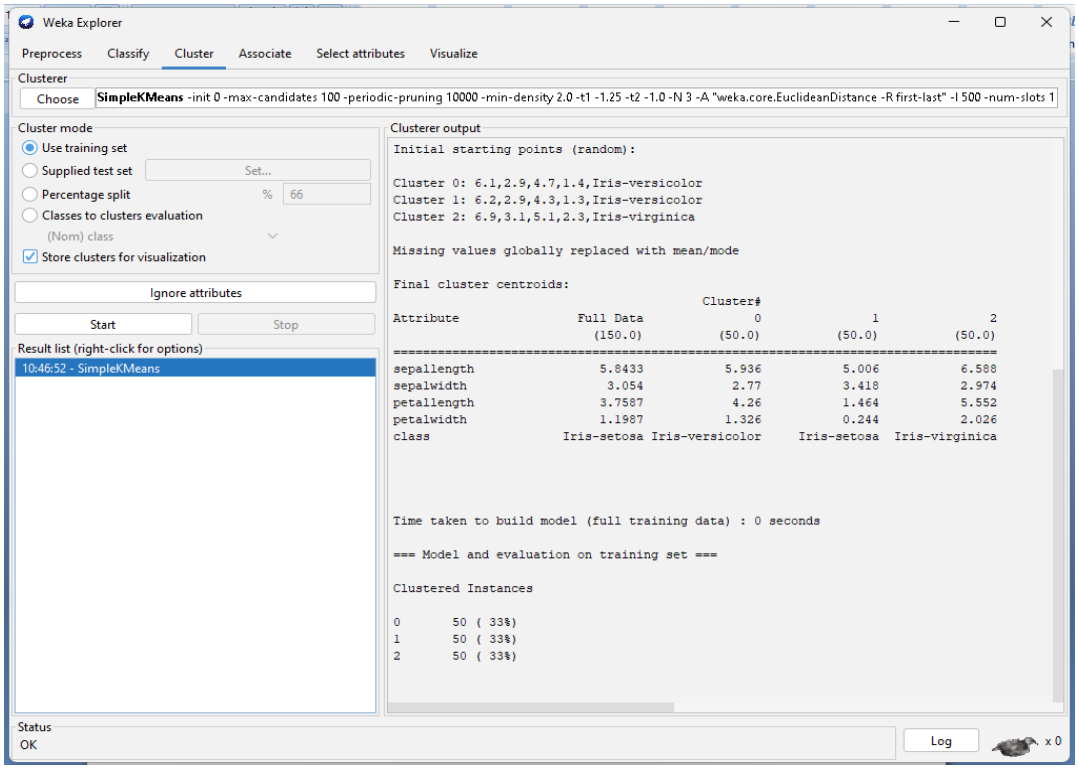


9. Now, click on “SimpleKMeans” and change the numClusters value to “3” to get 3 clusters

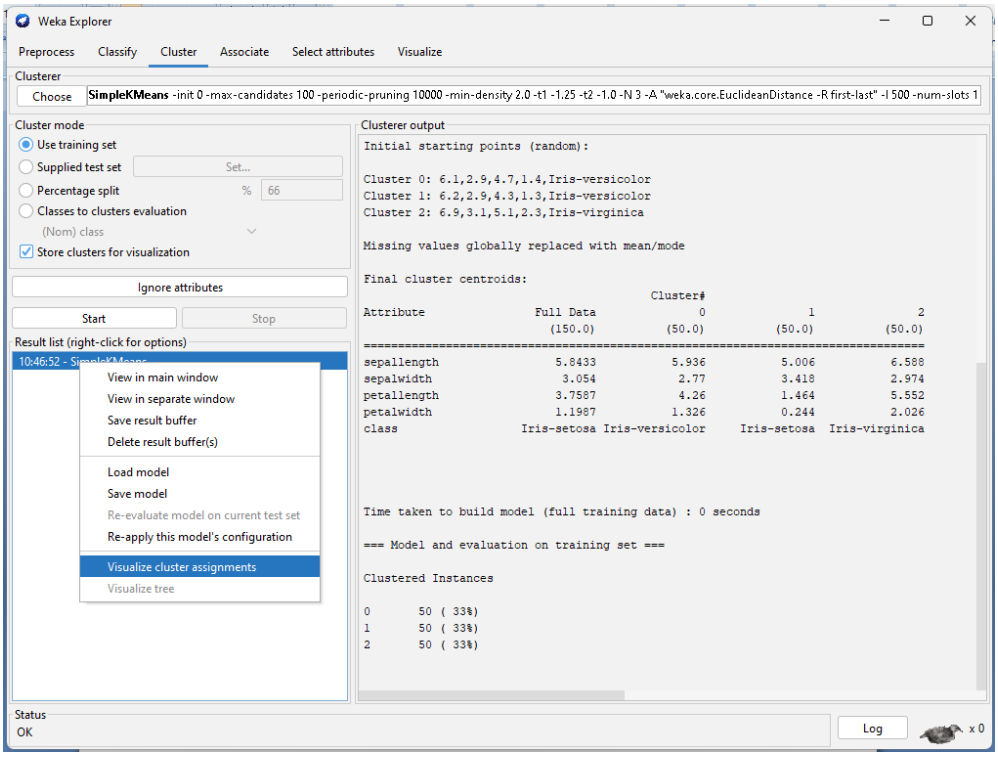


10. Click on “Start” to get the Cluster output

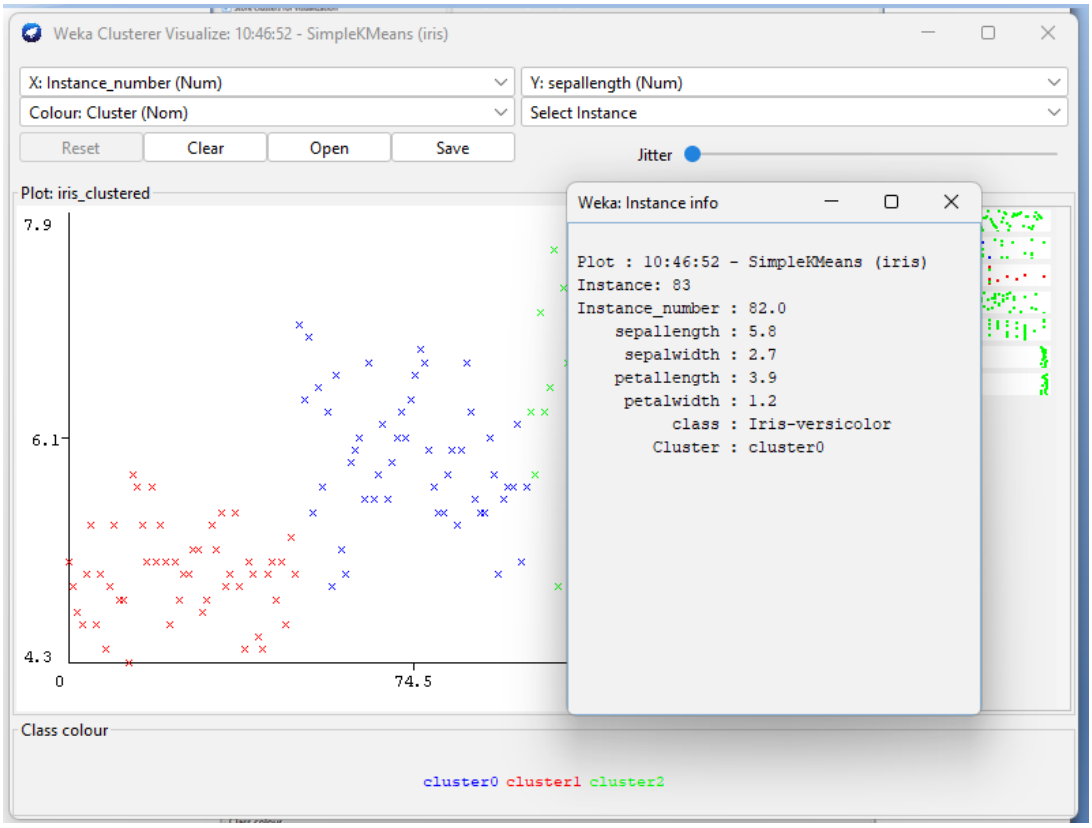




11. Now, right click on SimpleKMeans which is on Result list and then select the “Visualize cluster assignments”



12. Click on any instance on the graph to get the information of that instance



EXPERIMENT – 11

AIM:

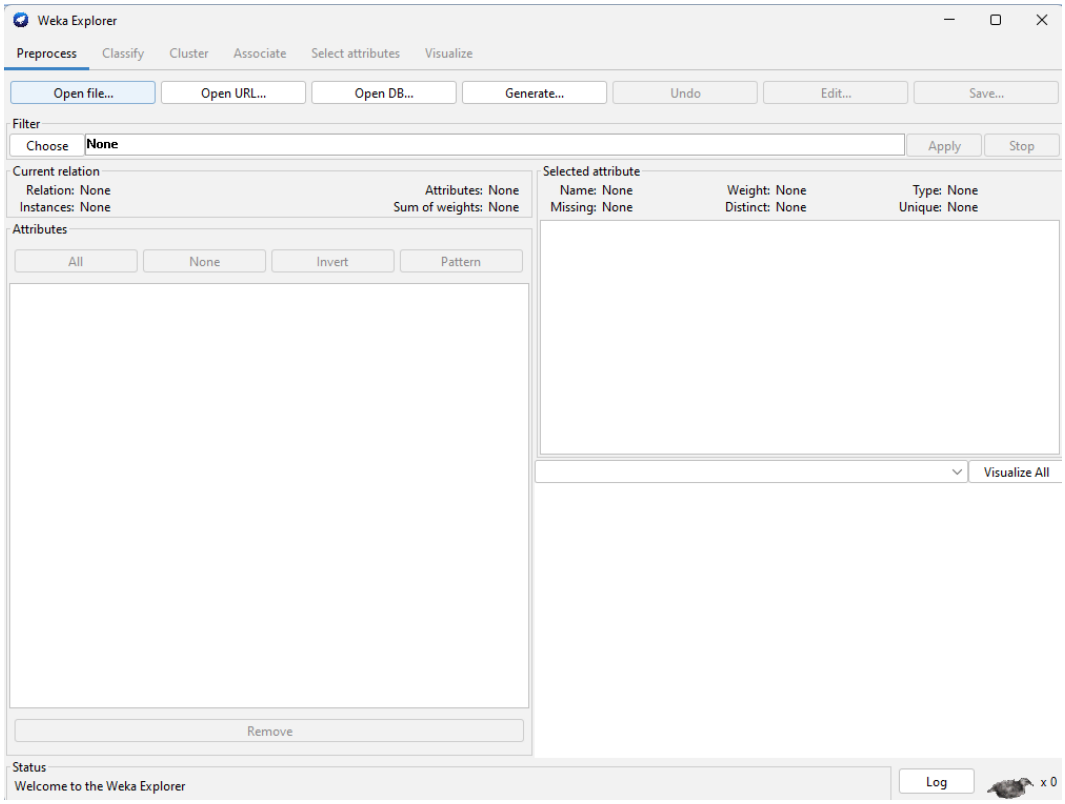
Utilize the Visualize option and tinker with the datapoints, available in it.

Description:

The visualization option in Weka tool allows users to create visual representations of their data through various charts and graphs. These visualizations can help users better understand their data, identify patterns, trends, and outliers. The tool provides a range of visualization techniques such as scatter plots, line charts, histogram, bar charts, and more. Users can customize the format, color, size, and other options of the visualizations to suit their needs.

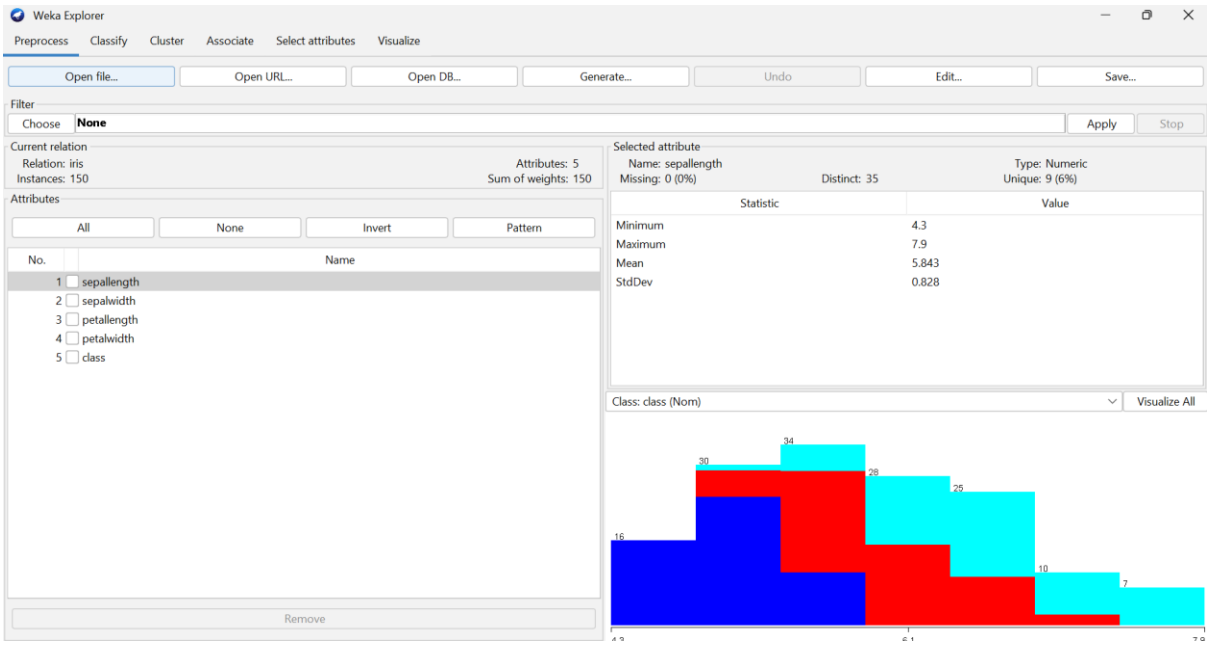
Procedure:

- open weka tool and select the explorer

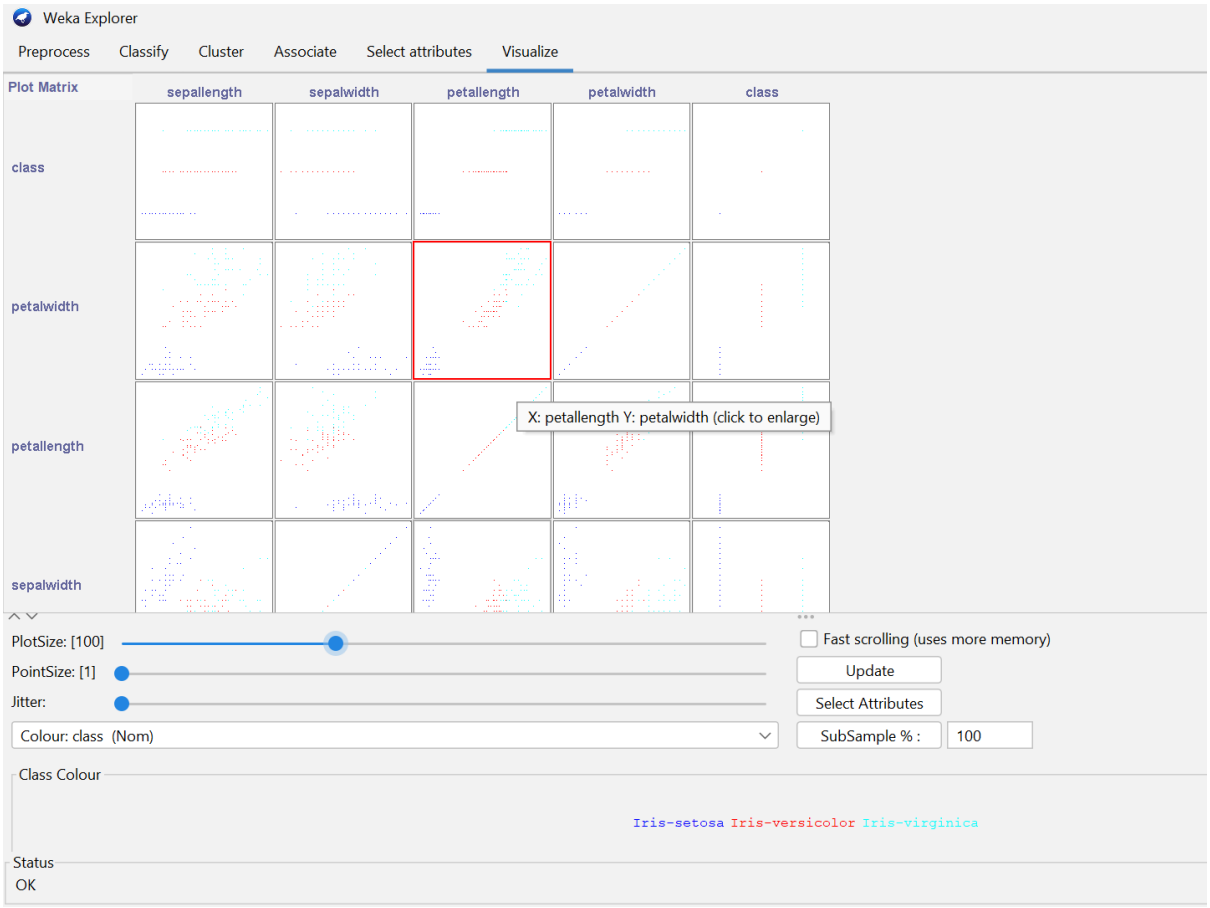


❖ load the iris dataset by selecting the “open file” option.

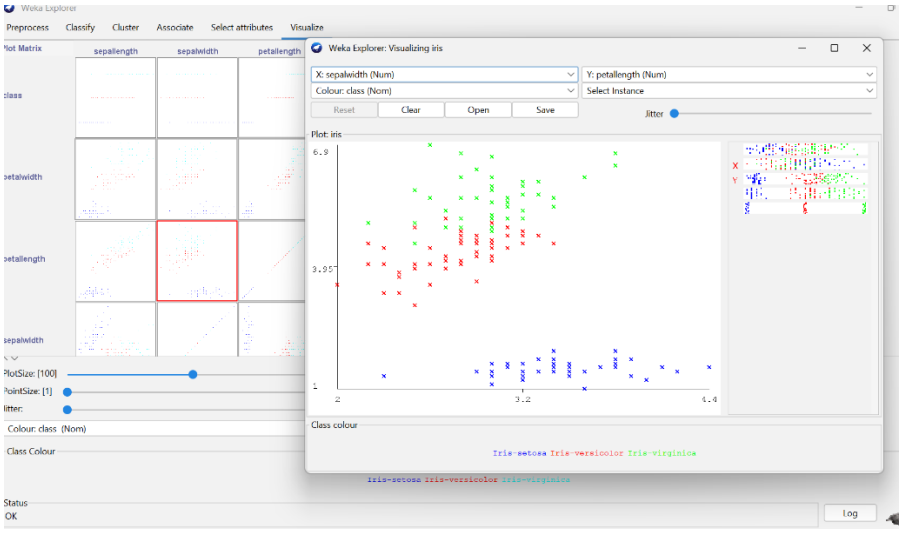
→upon successful opening



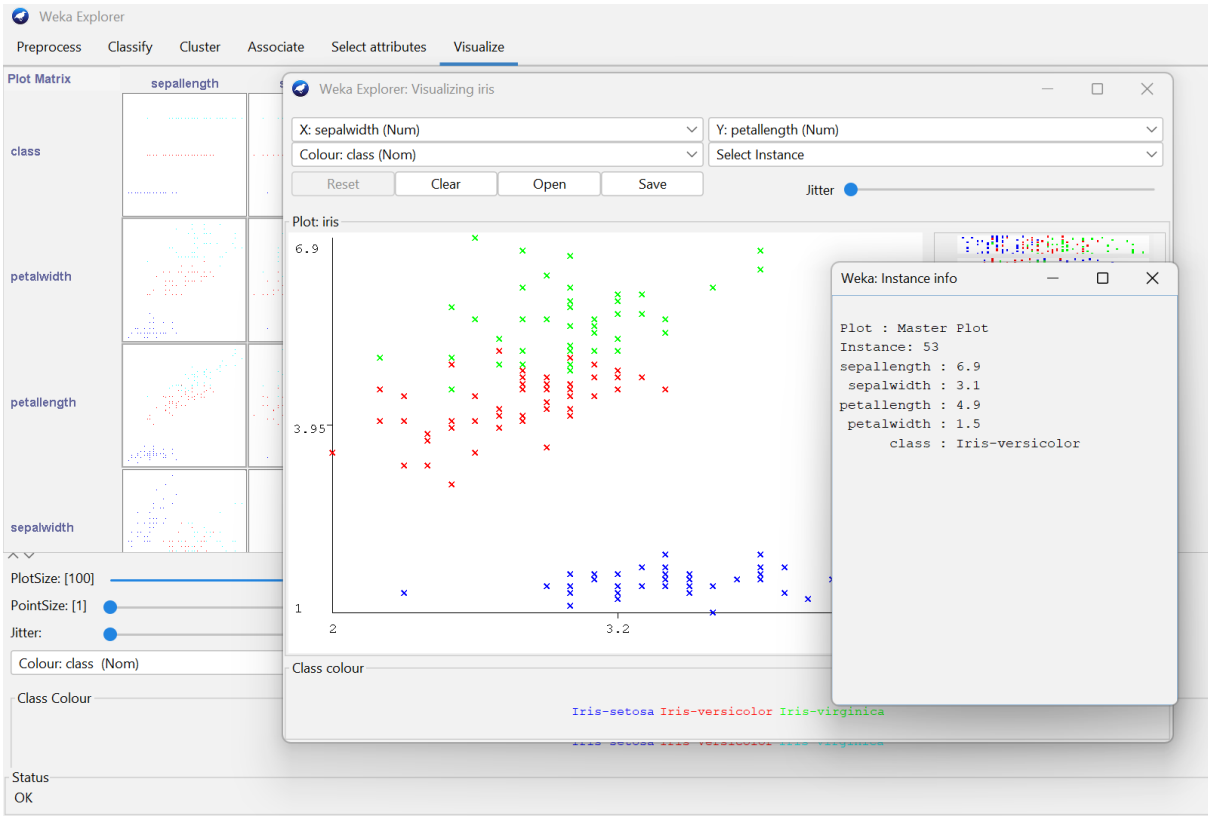
❖ Select visualize option from the toolbar.



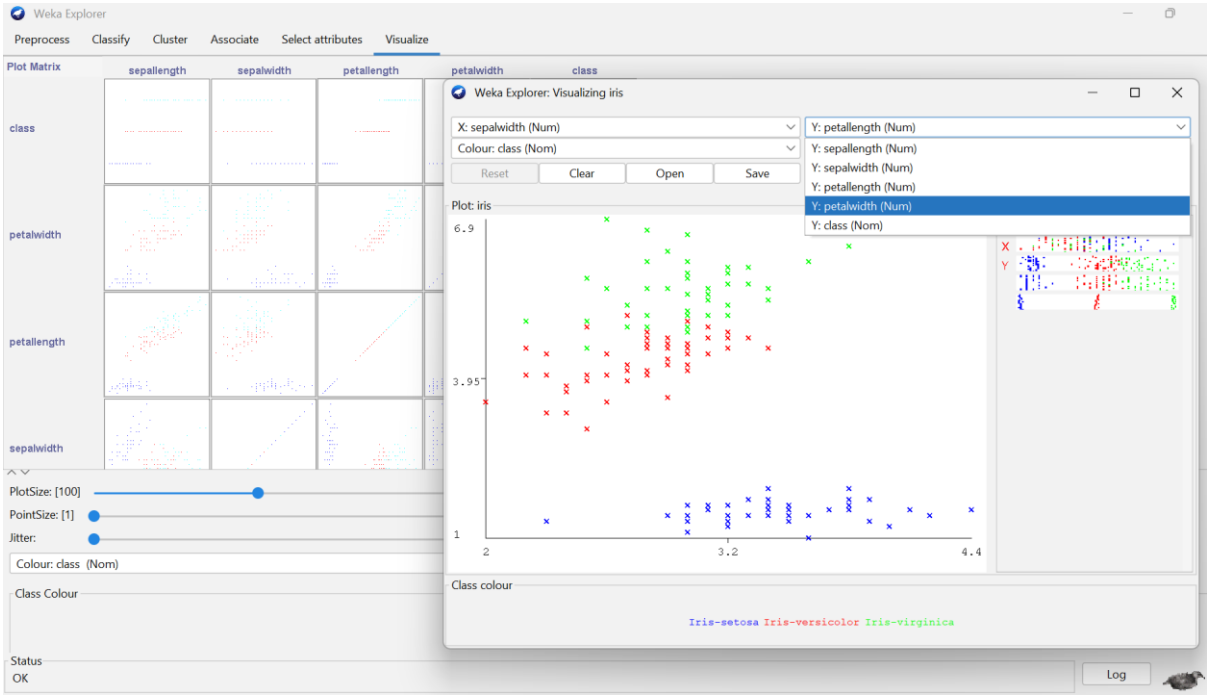
❖ open a selected box of your choice to enhance the view.



❖ select a single data point to dvelve into the details of it.



❖ Toggle the X or Y Co-ordinates by selecting the different features.



Experiment – 12

Aim: Apply Clustering technique on any dataset using Knowledge Flow.

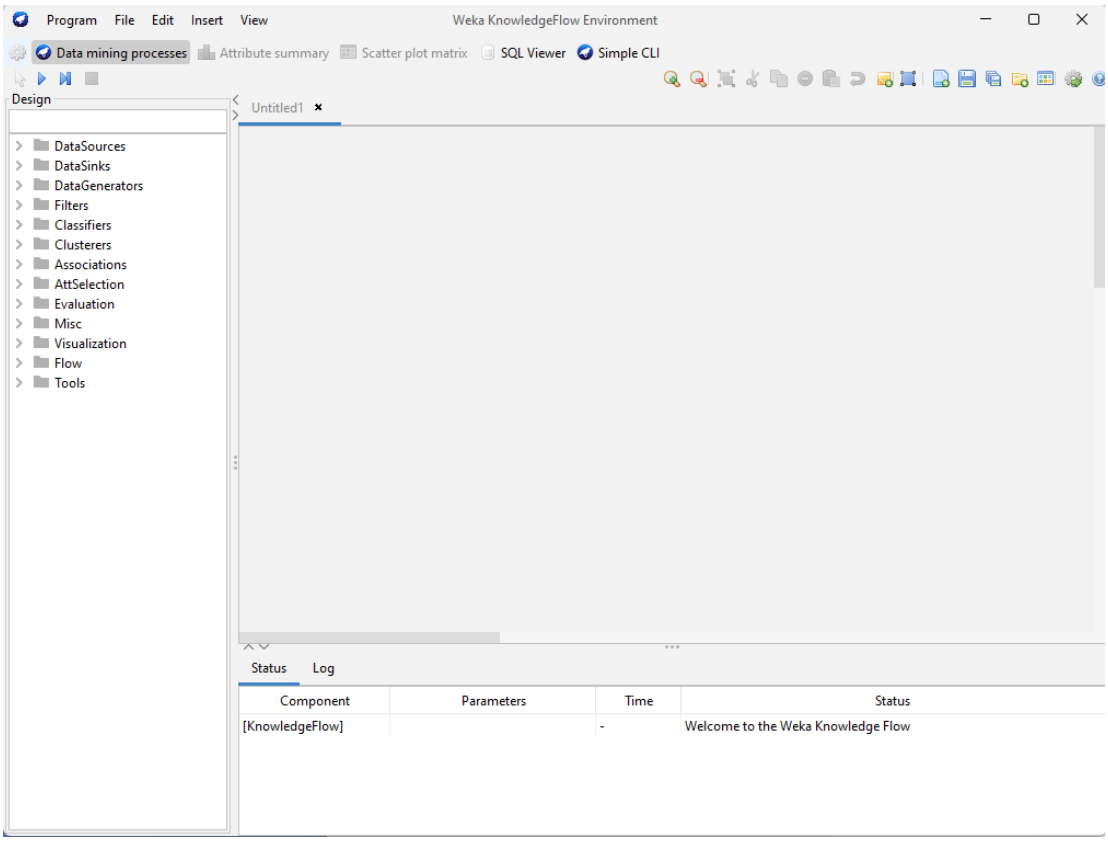
Description: Cluster analysis, also known as clustering, is a method of data mining that groups similar data points together. The goal of cluster analysis is to divide a dataset into groups (or clusters) such that the data points within each group are more similar to each other than to data points in other groups. This process is often used for exploratory data analysis and can help identify patterns or relationships within the data that may not be immediately obvious. There are many different algorithms used for cluster analysis, such as k-means, hierarchical clustering, and density-based clustering. The choice of algorithm will depend on the specific requirements of the analysis and the nature of the data being analysed.

Step by Step Process:

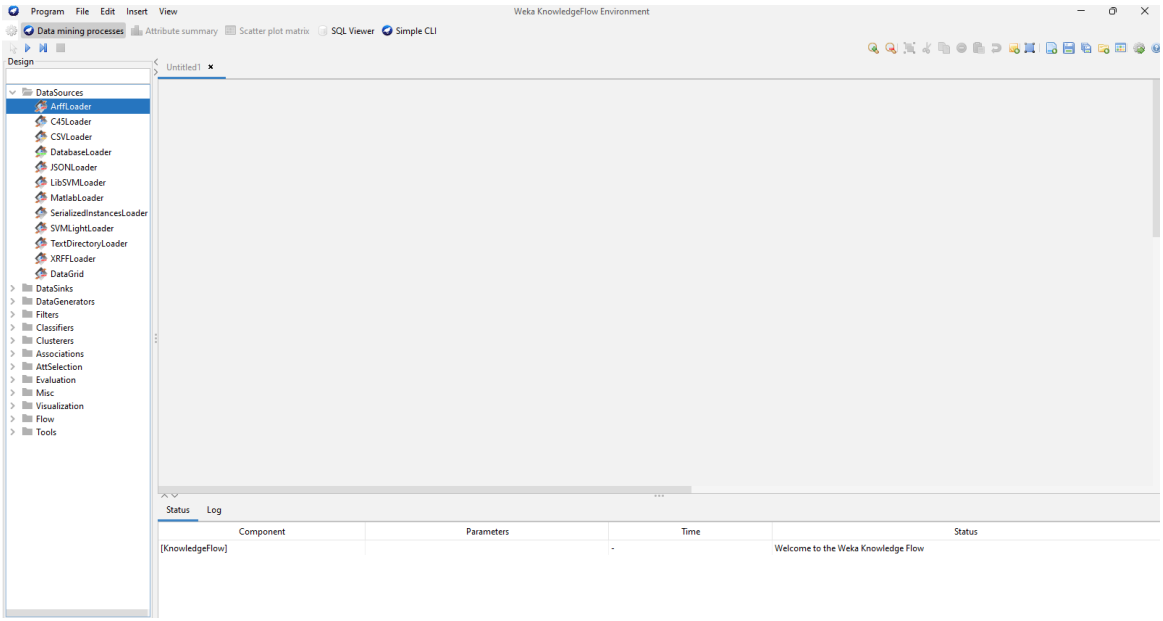
→*To Add Attribute*

1. Open WEKA Tool and navigate to Knowledge Flow.

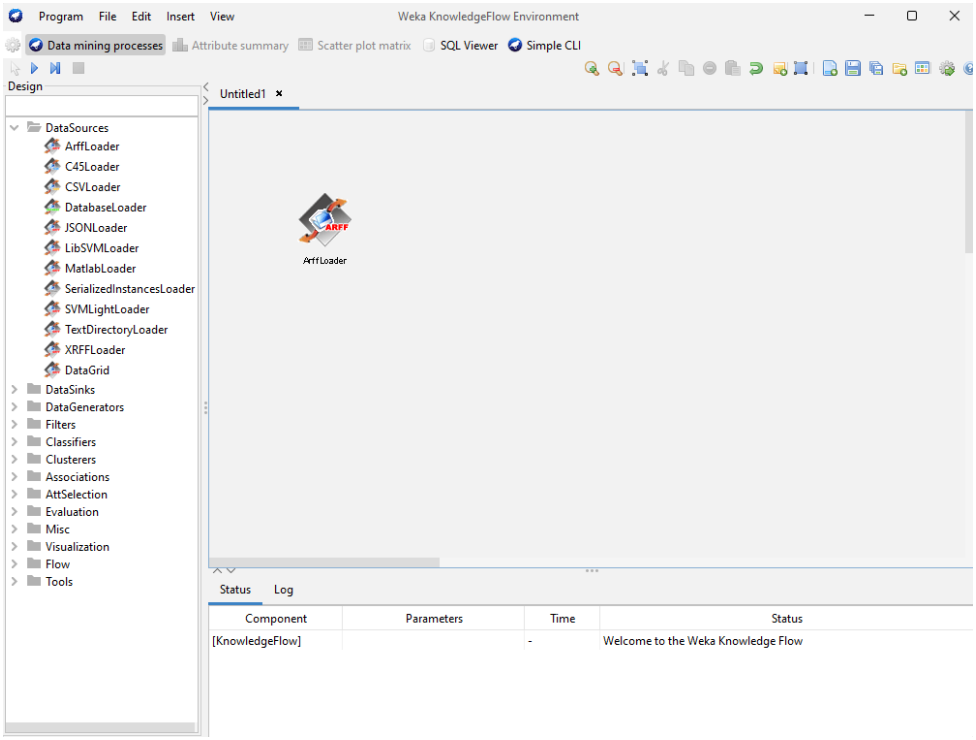
2. Knowledge Flow Environment Views



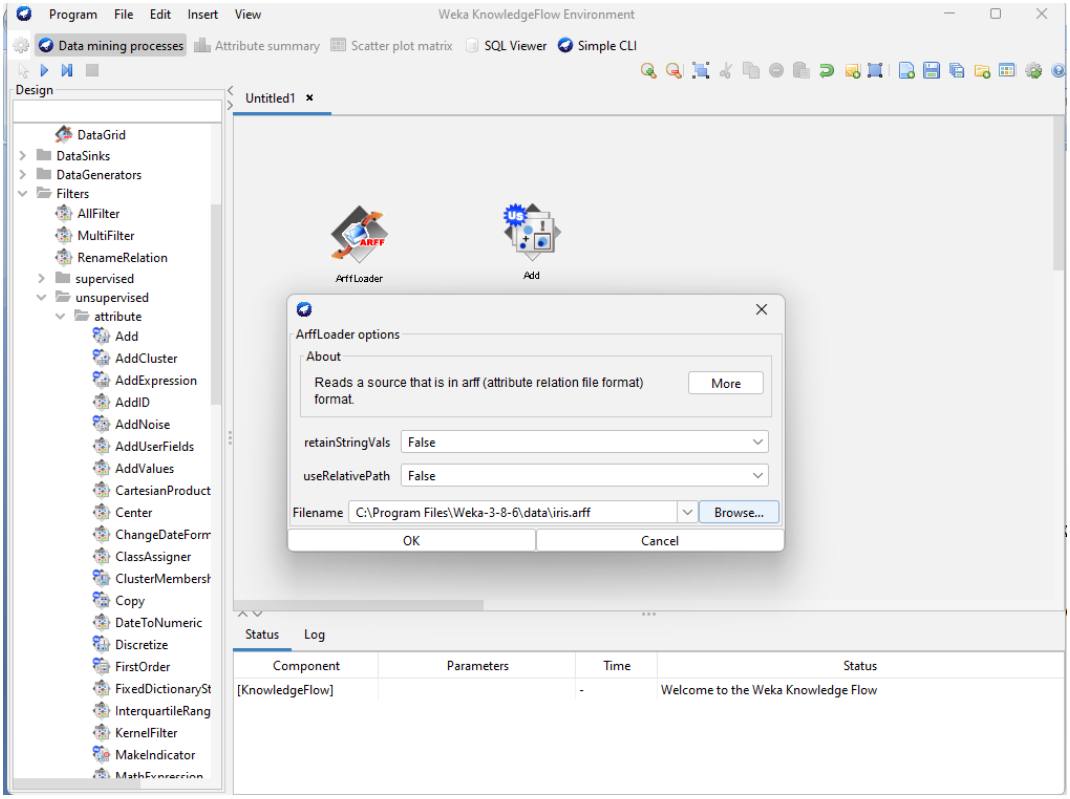
3. To create a model's Knowledge Flow, pick DataSources in the Design window. Expand data sources. To pick the dataset, use either ArffLoader or CSV Loader.



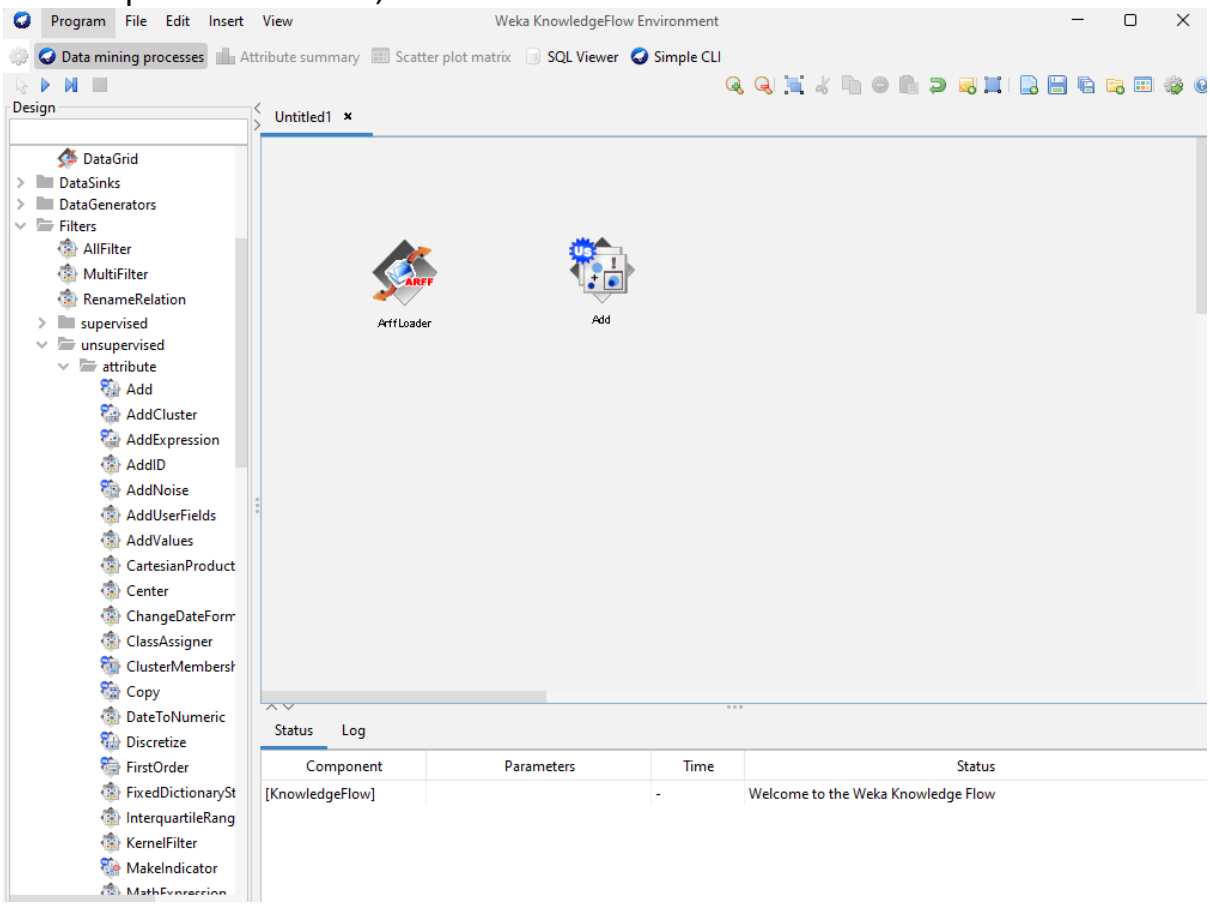
4. Select the Loader and set it in the Knowledge Flow by clicking on the canvas window.



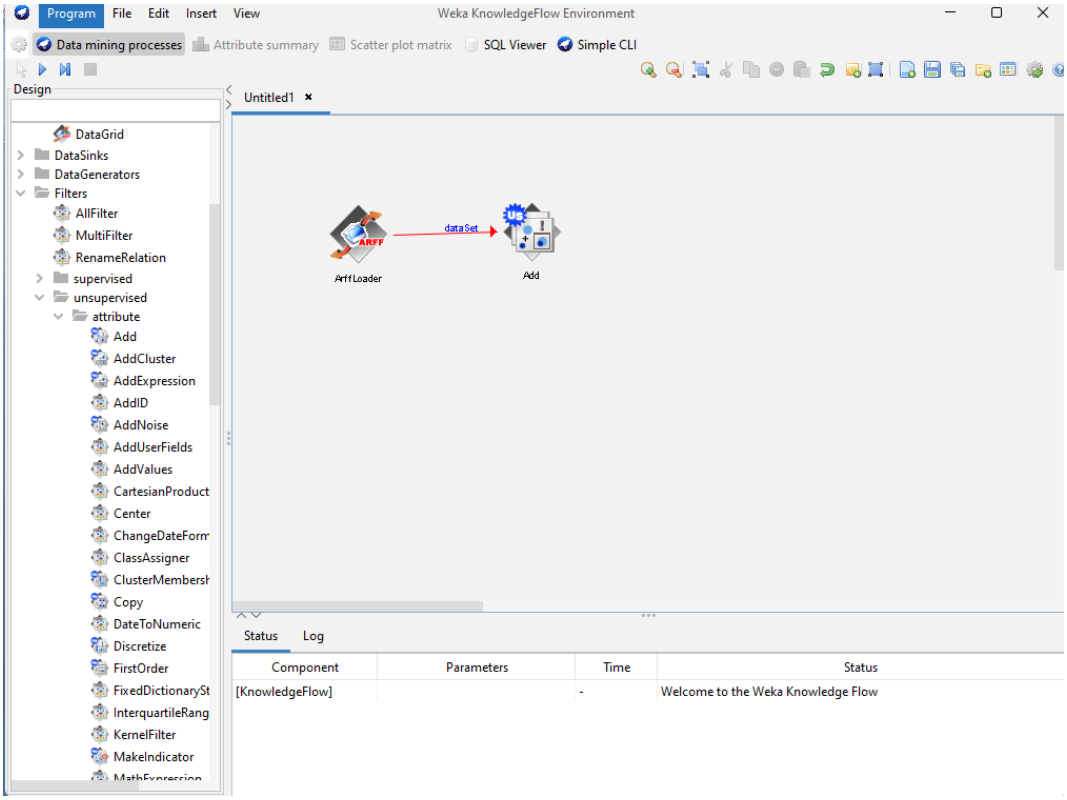
5. To load the dataset in ArffLoader, right-click on it, select Configure, and then pick the dataset.



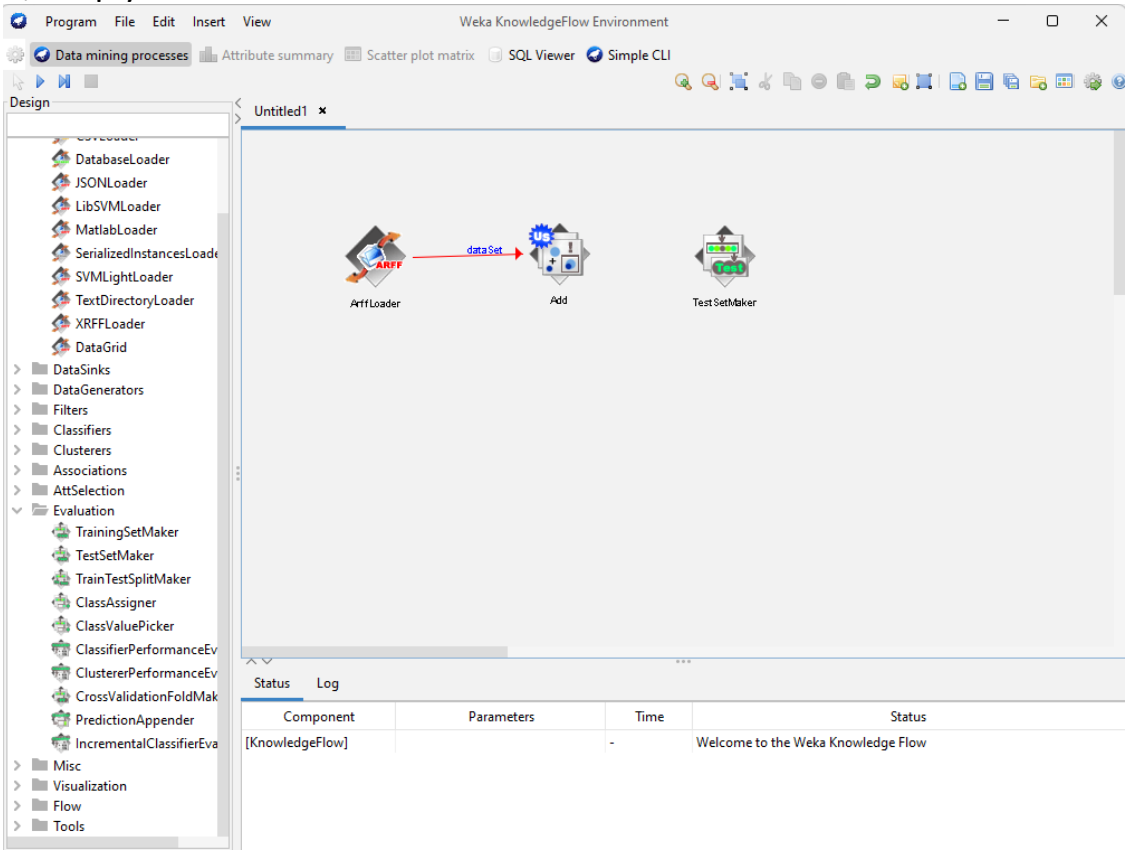
6. To add an attribute, navigate to Filters > Unsupervised > Attributes > Add. To position the Add, select it and then click on the canvas window.



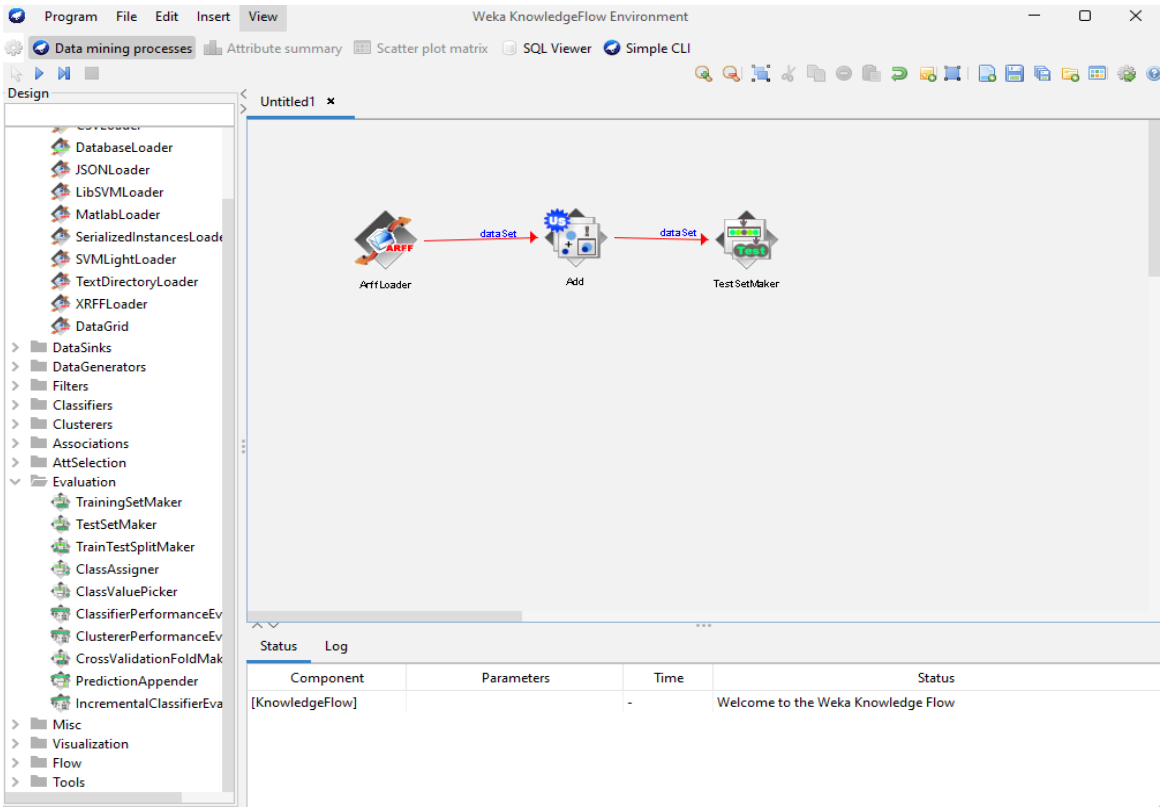
7. To link ArffLoader and Add, right-click on Arffoader>dataSet and choose Add.



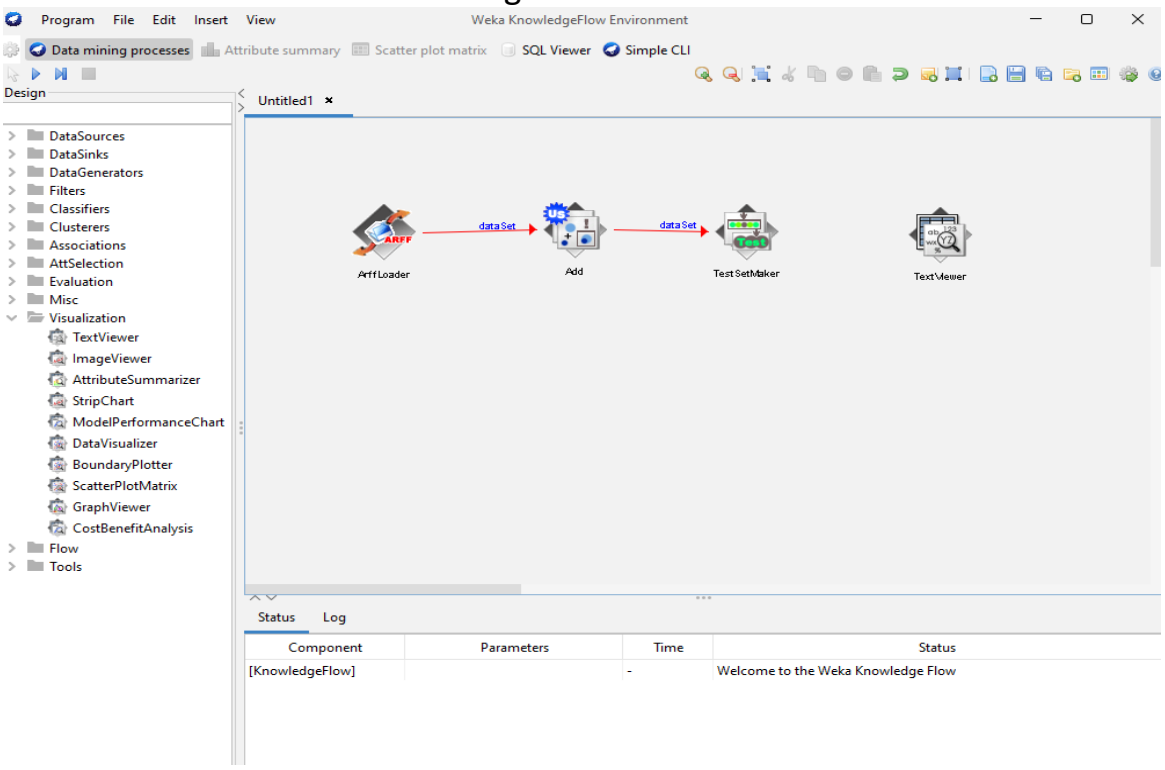
8. To evaluate the DataSet, choose Evaluation > TestSetMaker. To position it, simply click on the canvas window.



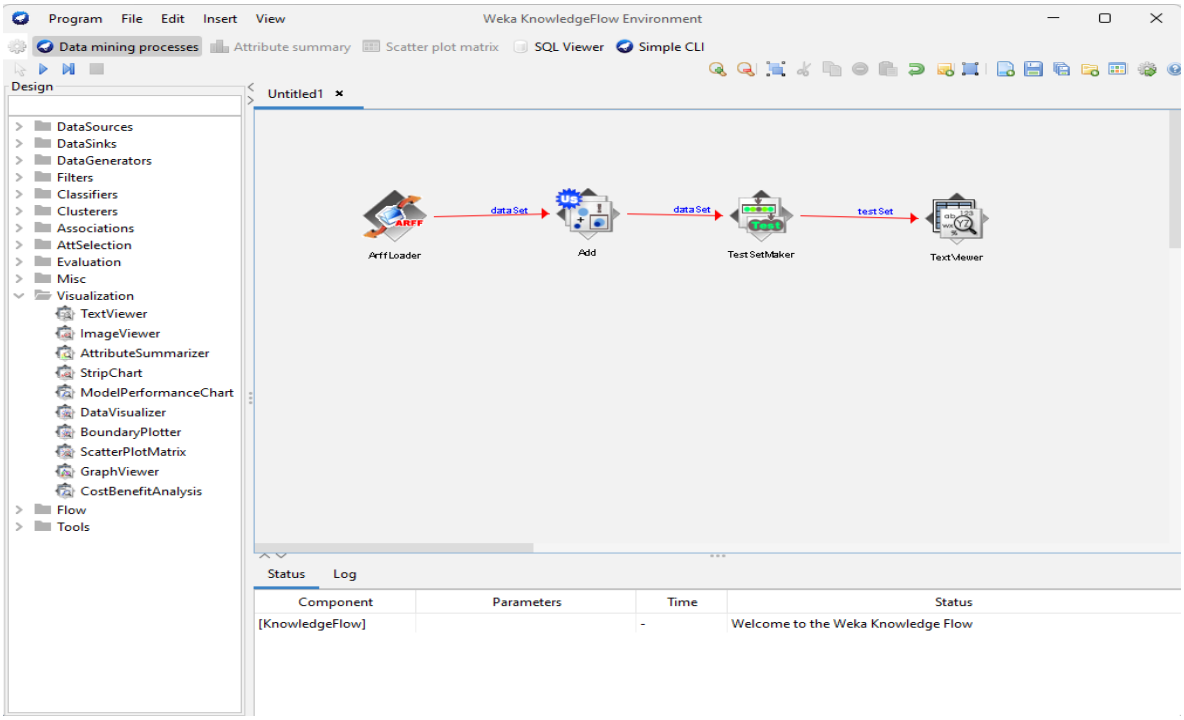
9. To link Add and TestSetMaker, right-click on Add>dataSet and then choose TestSetMaker.



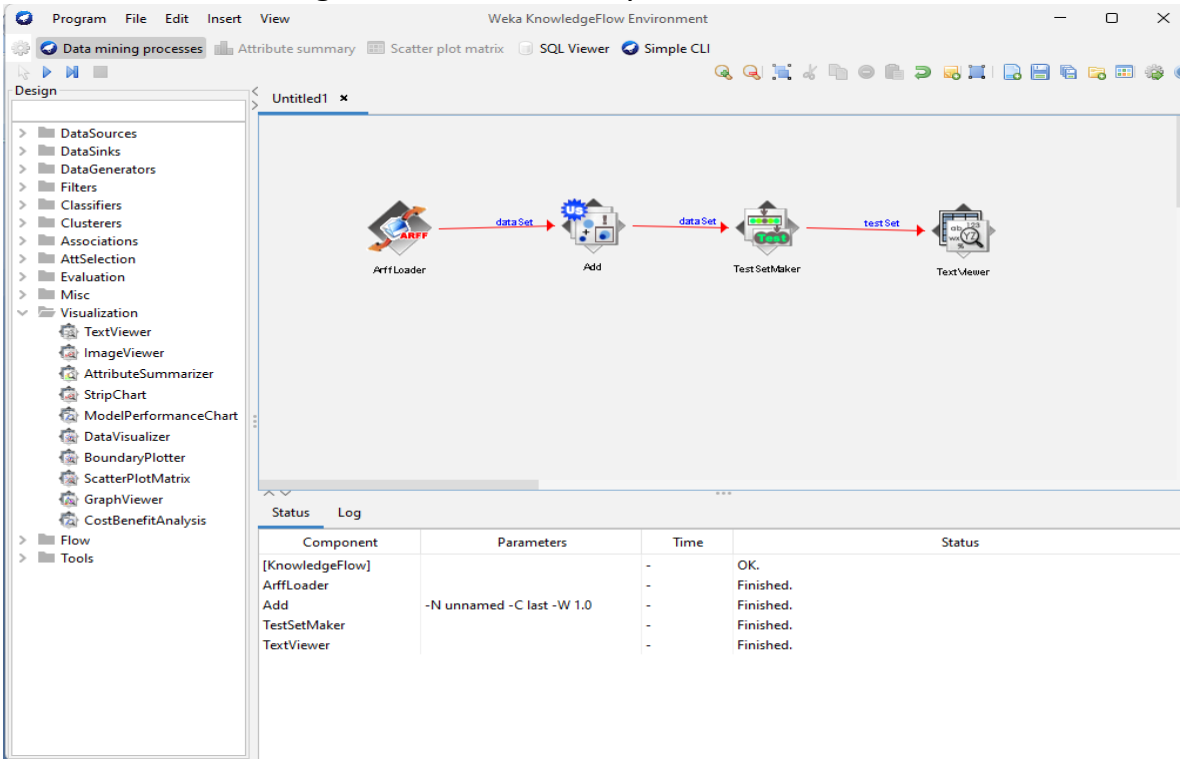
10. To view the findings, we must utilize Visualization. Choose Visualization>TextViewer and drag it onto the canvas window.



11. To create a link between TestSetMaker and TextViewer, right-click on TestSetMaker, then choose TestSet and drop it on TextViewer.



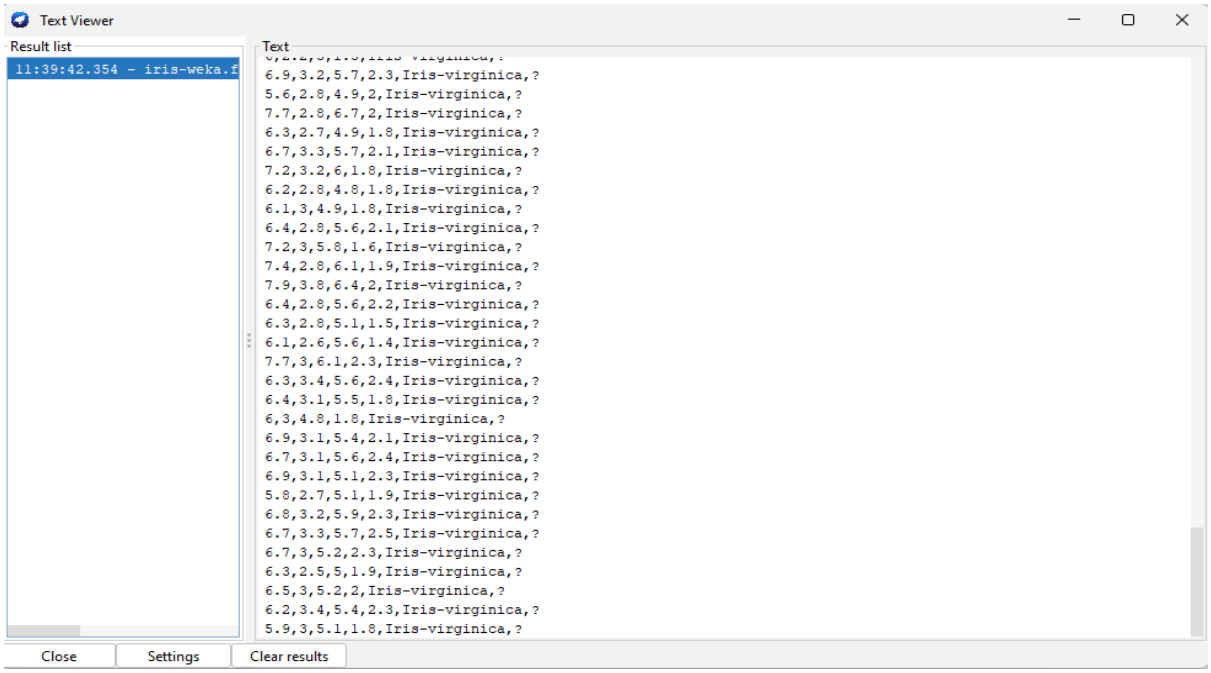
12. To start the Knowledge Flow, click the Play button in the Menu bar.



We can see the execution status in Status Window.

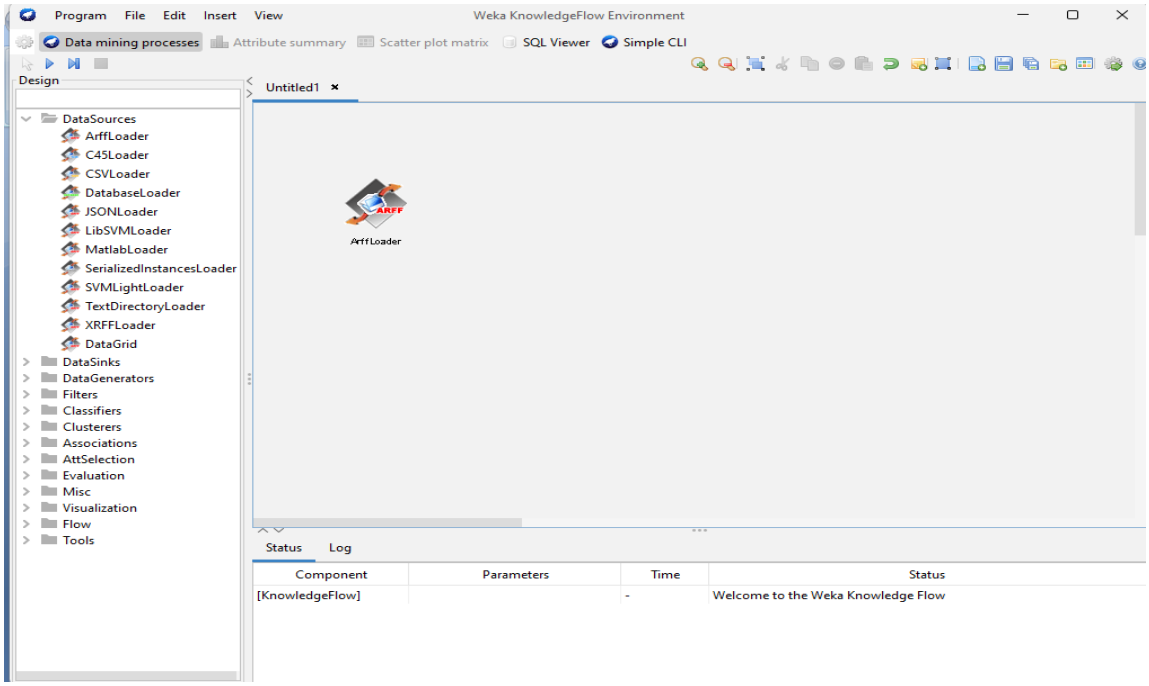
13. To view the results, right-click on TextViewer and select Show Results.

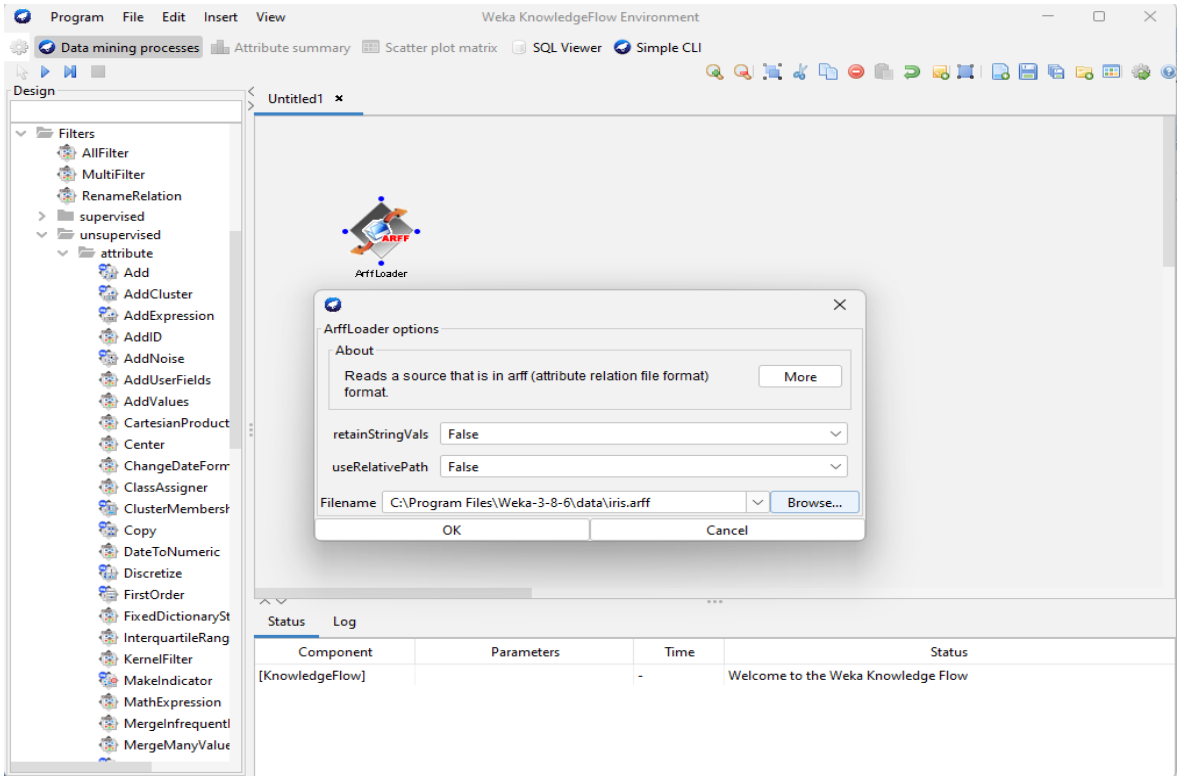
```
@relation iris-weka.filters.unsupervised.attribute.Add-Nunnamed-Clast-W1.0

@attribute sepallength numeric
@attribute sepalwidth numeric
@attribute petallength numeric
@attribute petalwidth numeric
@attribute class {Iris-setosa,Iris-versicolor,Iris-virginica}
@attribute unnamed numeric

@data
5.1,3.5,1.4,0.2,Iris-setosa,?
4.9,3,1.4,0.2,Iris-setosa,?
4.7,3.2,1.3,0.2,Iris-setosa,?
```

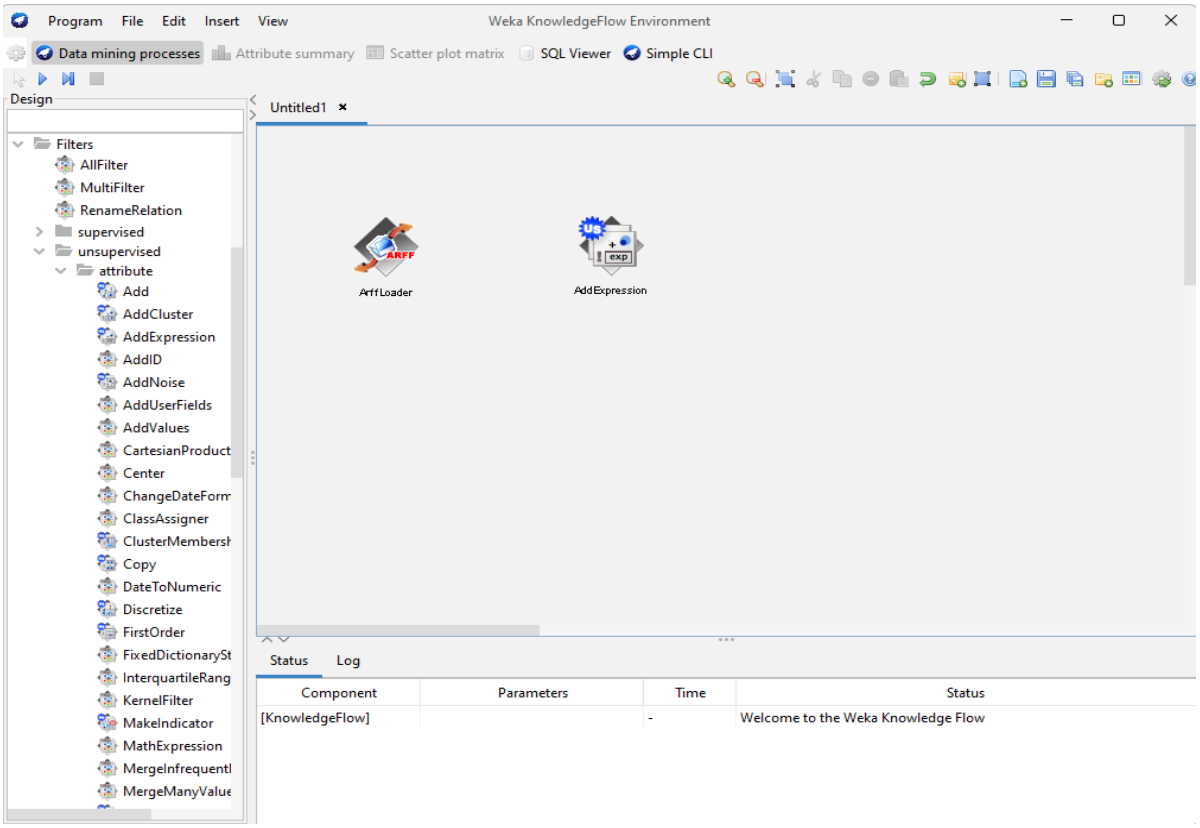


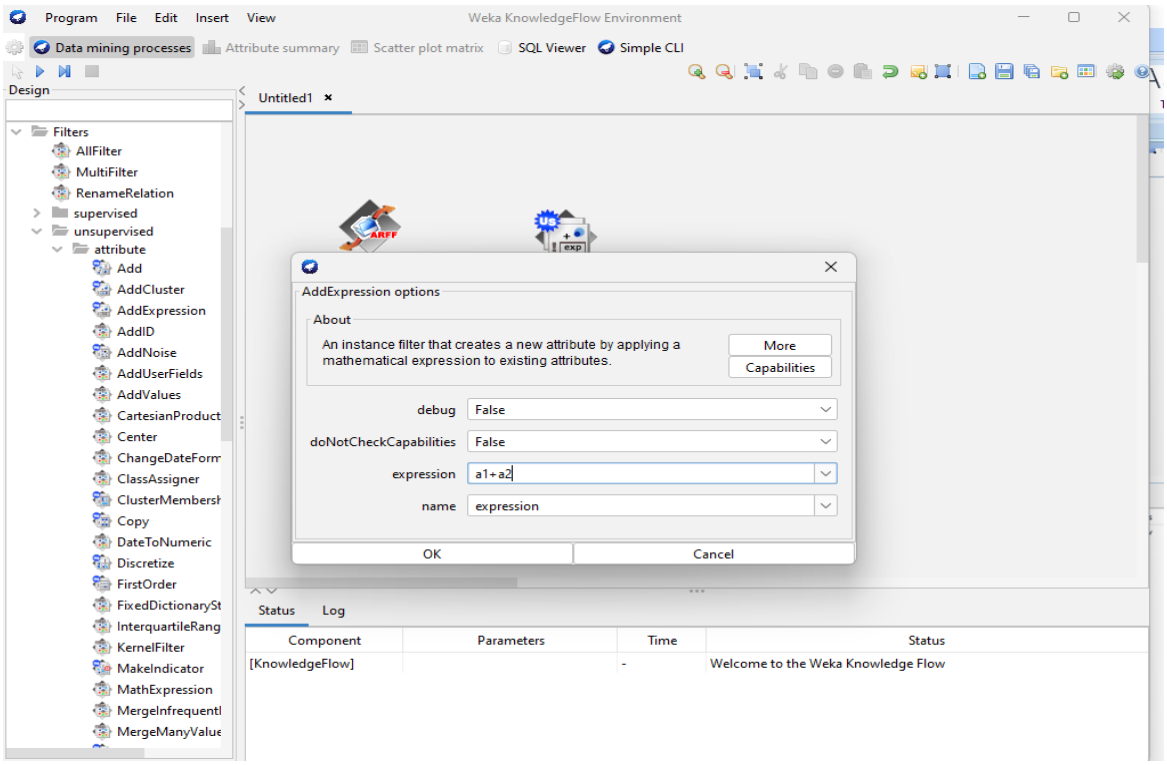
The ? symbol symbolizes the un-inserted values in the newly introduced attribute.

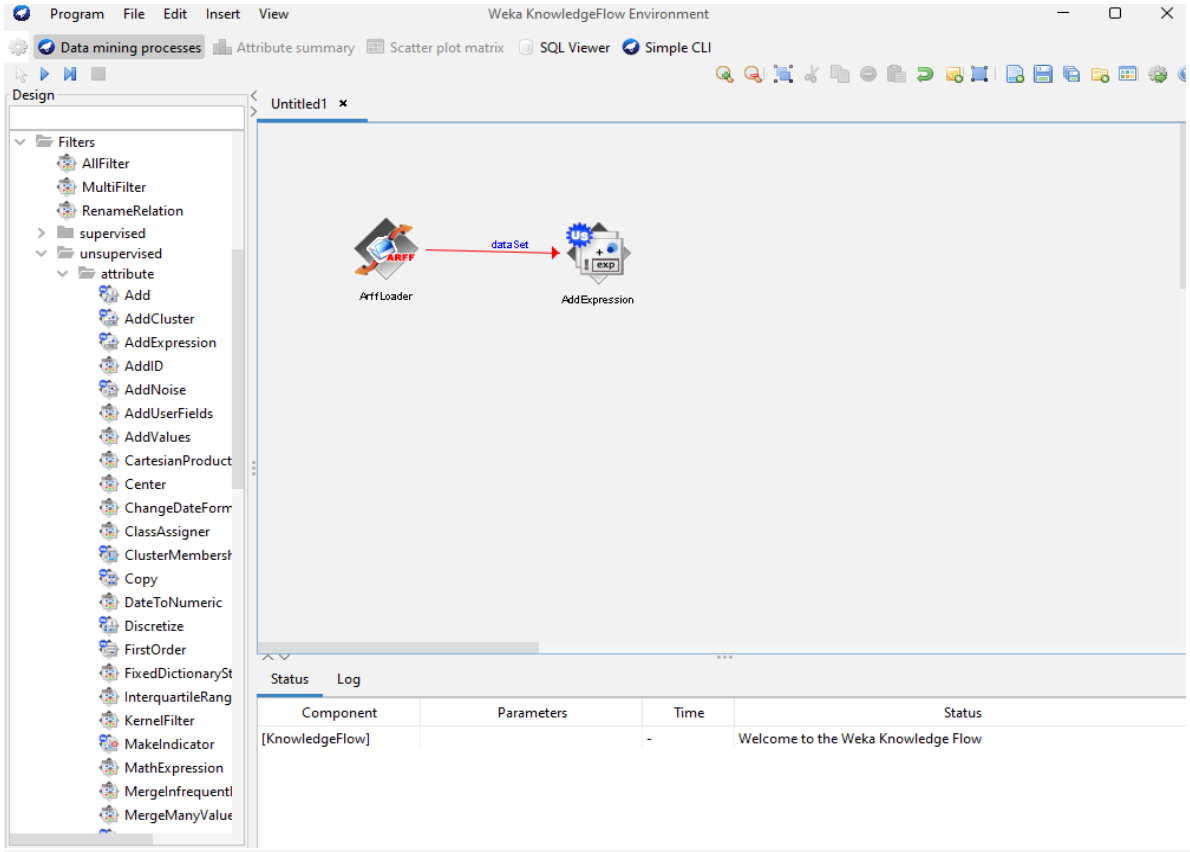
→ *To Add Expression:*

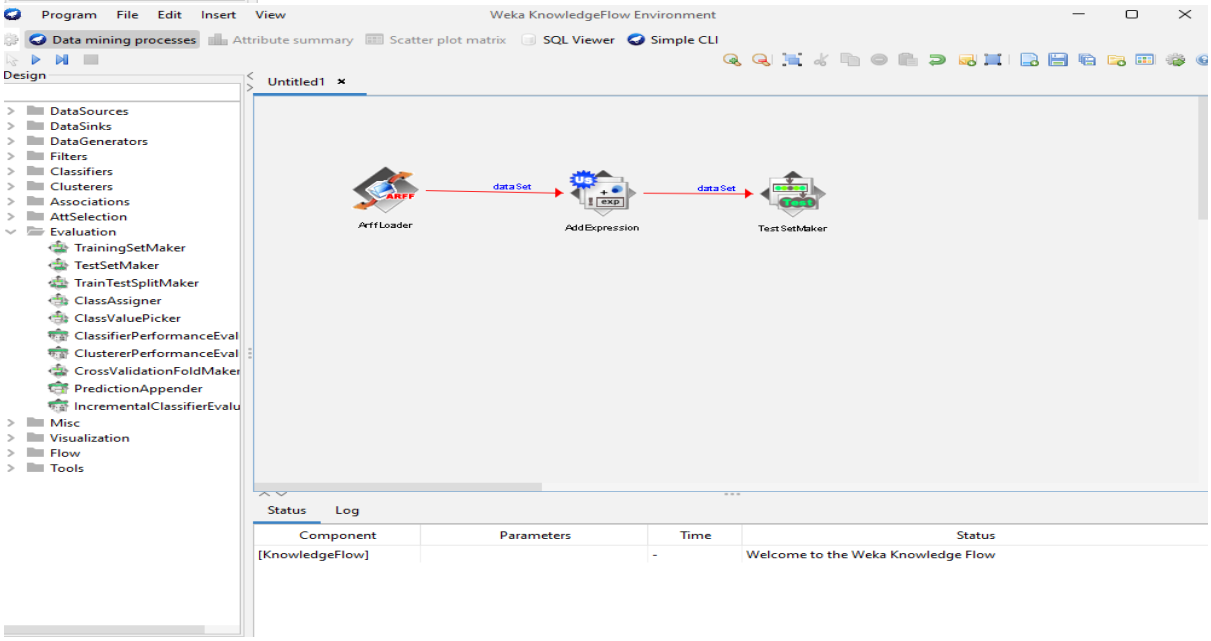

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Untitled1
DataSources
ArffLoader
C45Loader
CSVLoader
DatabaseLoader
JSONLoader
LibSVMLoader
MatlabLoader
SerializedInstancesLoader
SVMLightLoader
TextDirectoryLoader
XRFFLoader
DataGrid
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
Misc
Visualization
Flow
Tools
ArffLoader
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

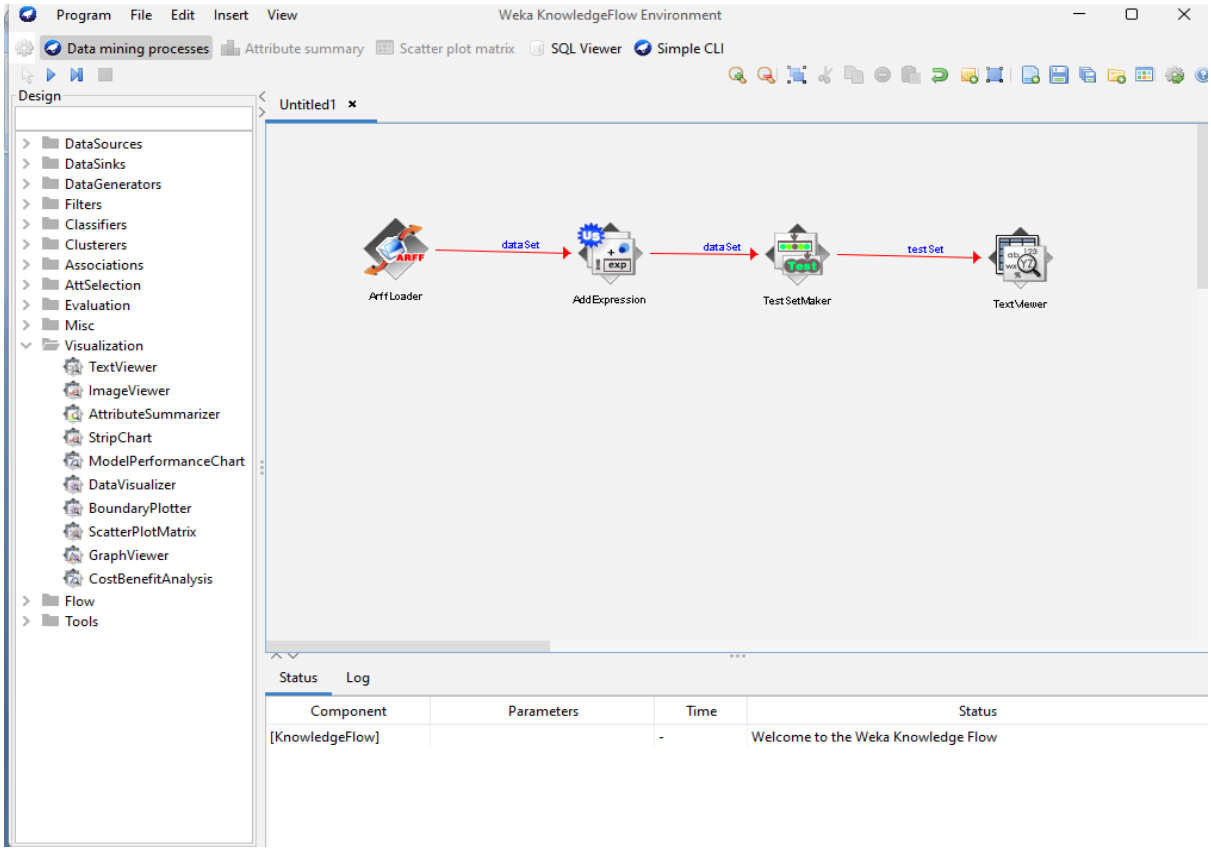

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Untitled1
Filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
ArffLoader
ArffLoader options
About
Reads a source that is in arff (attribute relation file format) format.
More
retainStringVals
False
useRelativePath
False
Filename
C:\Program Files\Weka-3-8-6\data\iris.arff
Browse...
OK
Cancel
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

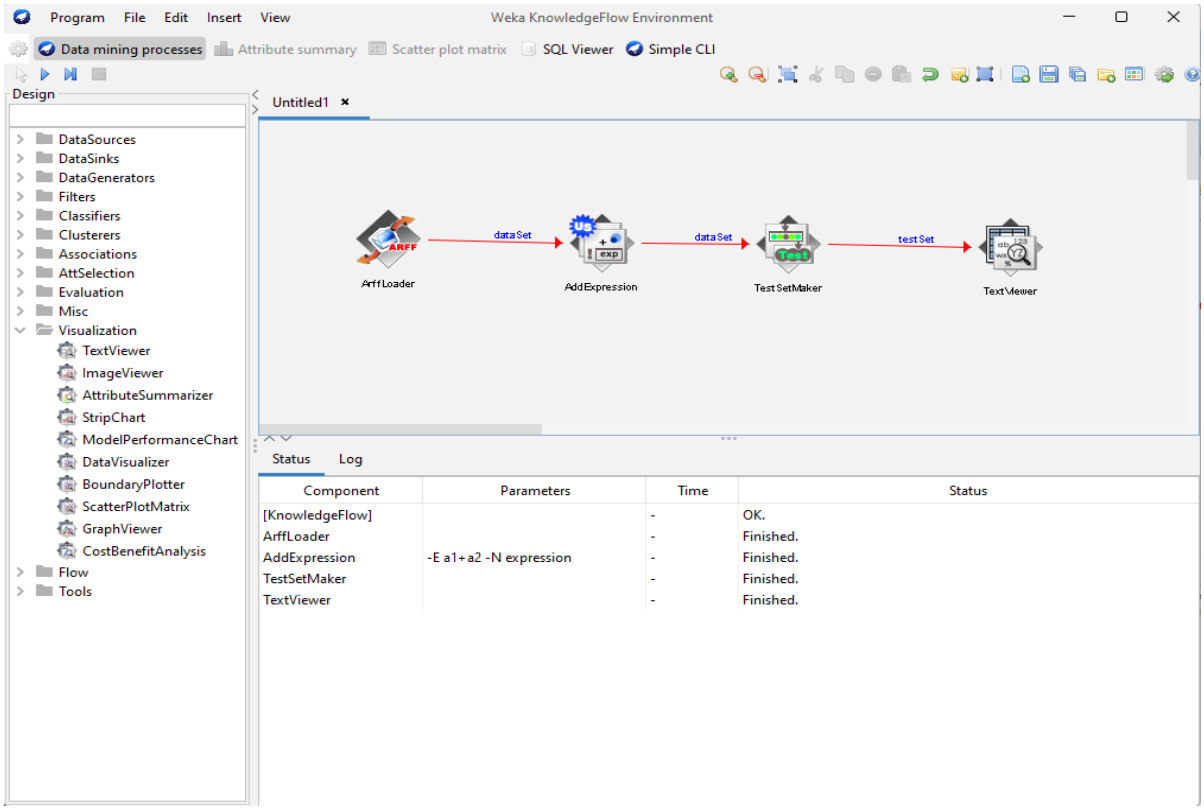

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
Untitled1
ArffLoader
AddExpression
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
Untitled1
AddExpression options
About
An instance filter that creates a new attribute by applying a mathematical expression to existing attributes.
More
Capabilities
debug
False
doNotCheckCapabilities
False
expression
a1+a2
name
expression
OK
Cancel
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
Untitled1
ARFF
dataSet
exp
ArffLoader
AddExpression
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

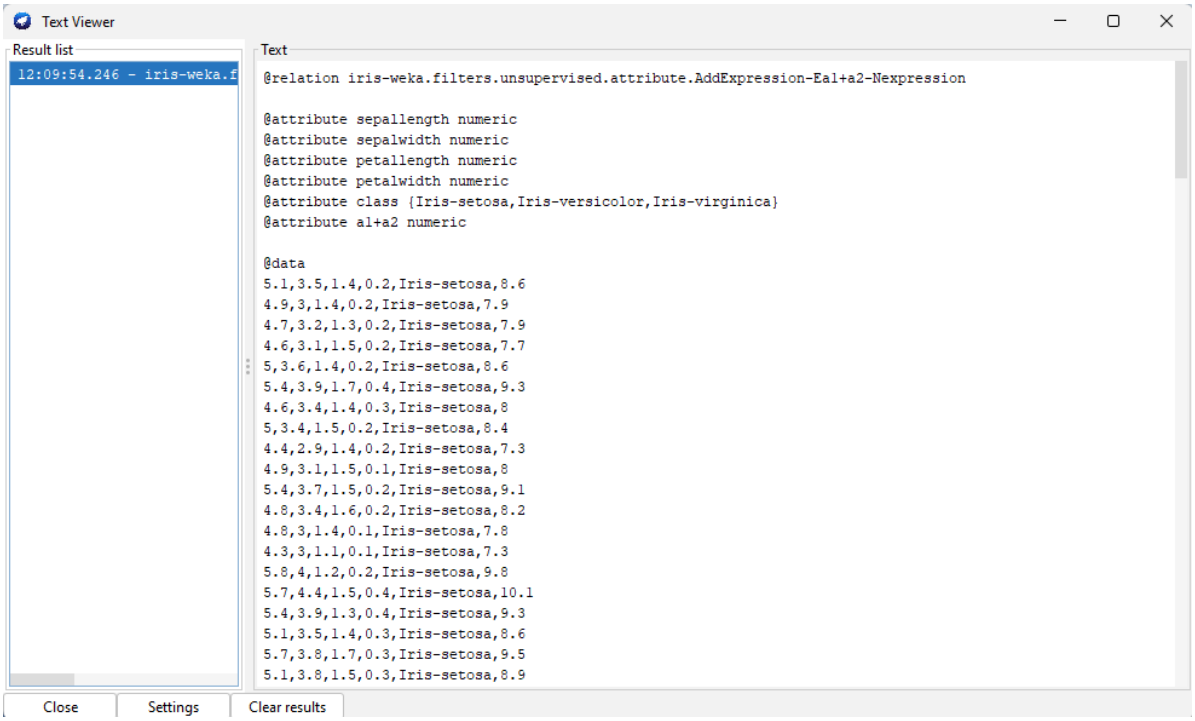

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
TrainingSetMaker
TestSetMaker
TrainTestSplitMaker
ClassAssigner
ClassValuePicker
ClassifierPerformanceEval
ClustererPerformanceEval
CrossValidationFoldMaker
PredictionAppender
IncrementalClassifierEvalu
Misc
Visualization
Flow
Tools
Untitled1
ARFF
dataSet
exp
ArffLoader
AddExpression
TestSetMaker
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
Misc
Visualization
TextViewer
ImageViewer
AttributeSummarizer
StripChart
ModelPerformanceChart
DataVisualizer
BoundaryPlotter
ScatterPlotMatrix
GraphViewer
CostBenefitAnalysis
Flow
Tools
Untitled1
ArffLoader
data Set
AddExpression
data Set
TestSetMaker
test Set
TextViewer
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

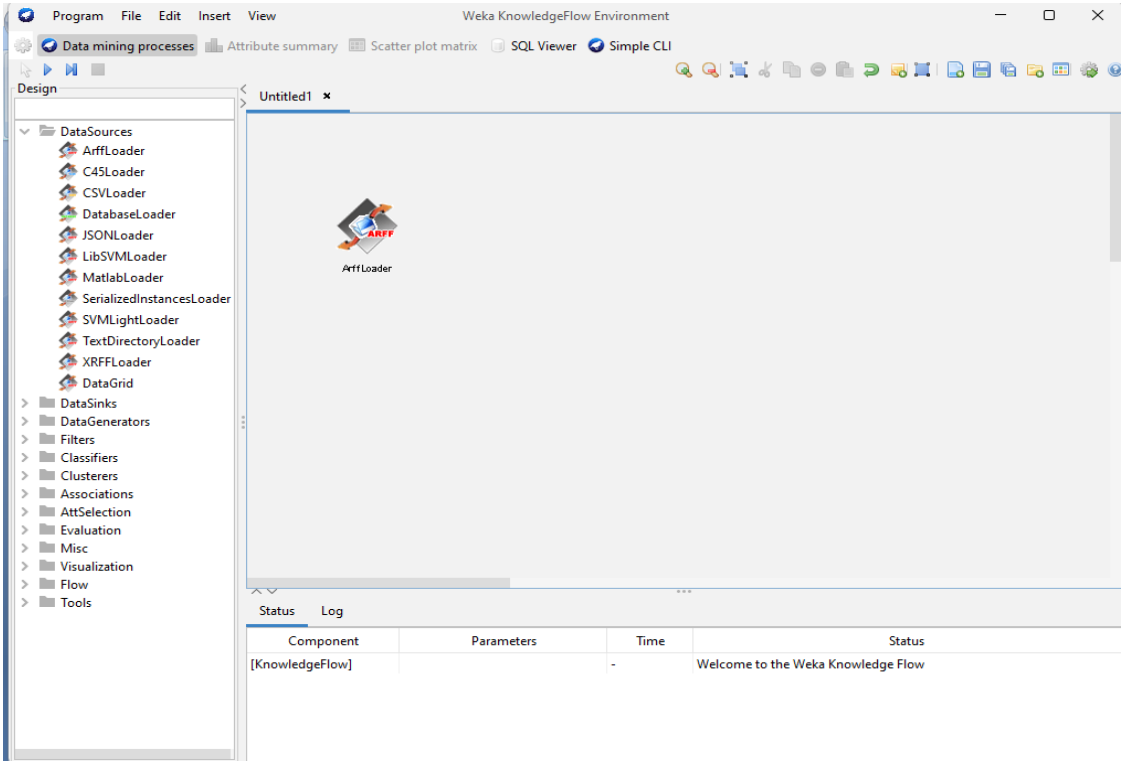

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
Misc
Visualization
TextViewer
ImageViewer
AttributeSummarizer
StripChart
ModelPerformanceChart
DataVisualizer
BoundaryPlotter
ScatterPlotMatrix
GraphViewer
CostBenefitAnalysis
Flow
Tools
Untitled1
ArffLoader
data Set
AddExpression
data Set
TestSetMaker
test Set
TextViewer
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
OK.
ArffLoader
-
Finished.
AddExpression
-E a1+a2 -N expression
-
Finished.
TestSetMaker
-
Finished.
TextViewer
-
Finished.

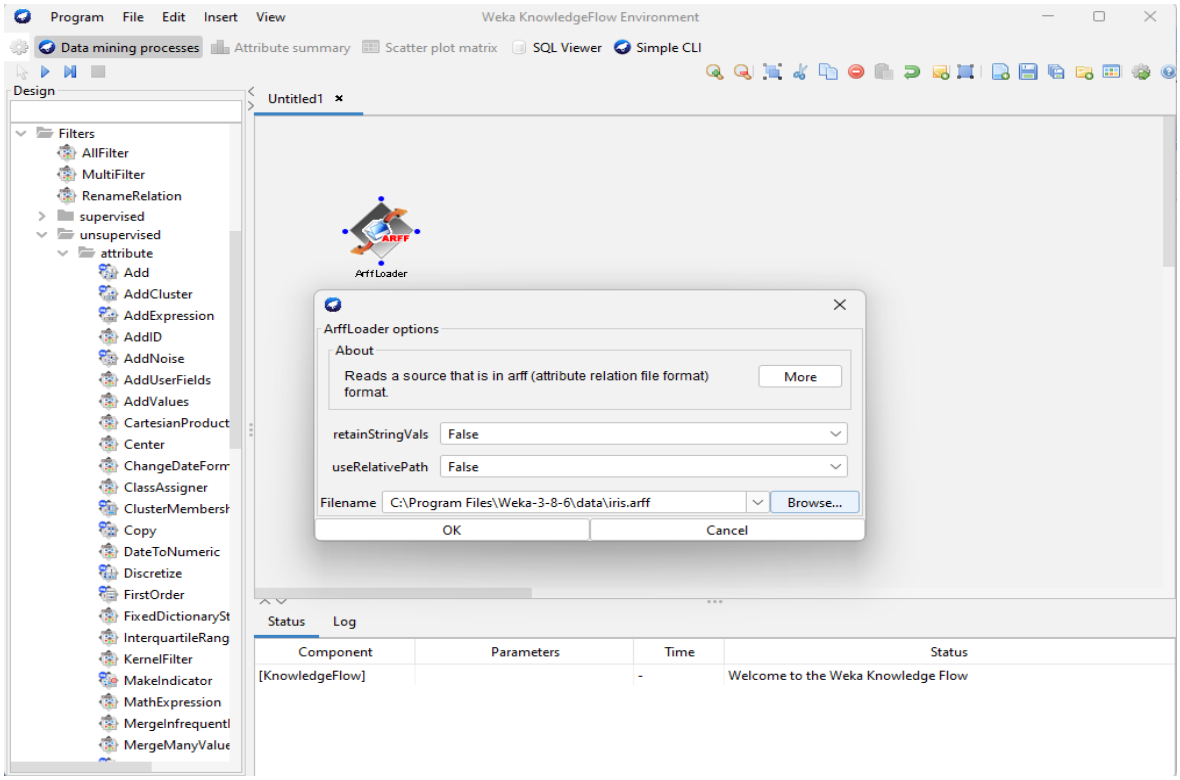
Result :

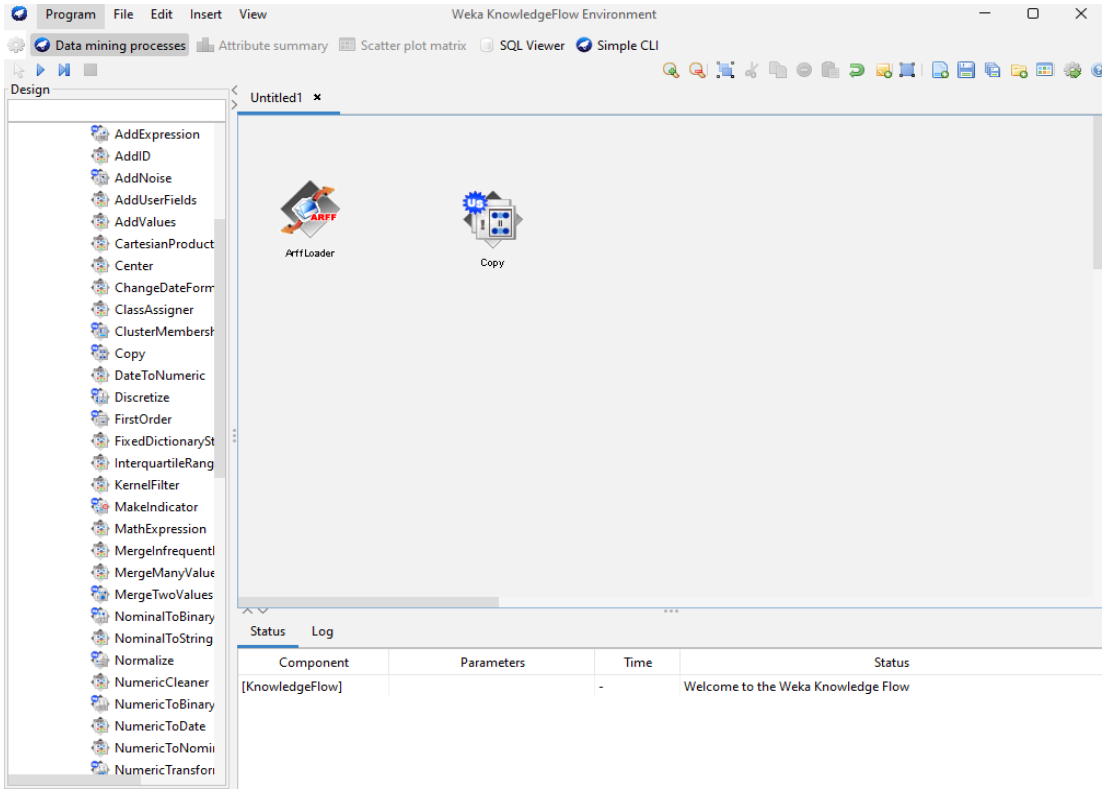
```
@relation iris-weka.filters.unsupervised.attribute.AddExpression-Ea1+a2-Nexpression

@attribute sepallength numeric
@attribute sepalwidth numeric
@attribute petallength numeric
@attribute petalwidth numeric
@attribute class {Iris-setosa,Iris-versicolor,Iris-virginica}
@attribute a1+a2 numeric
```

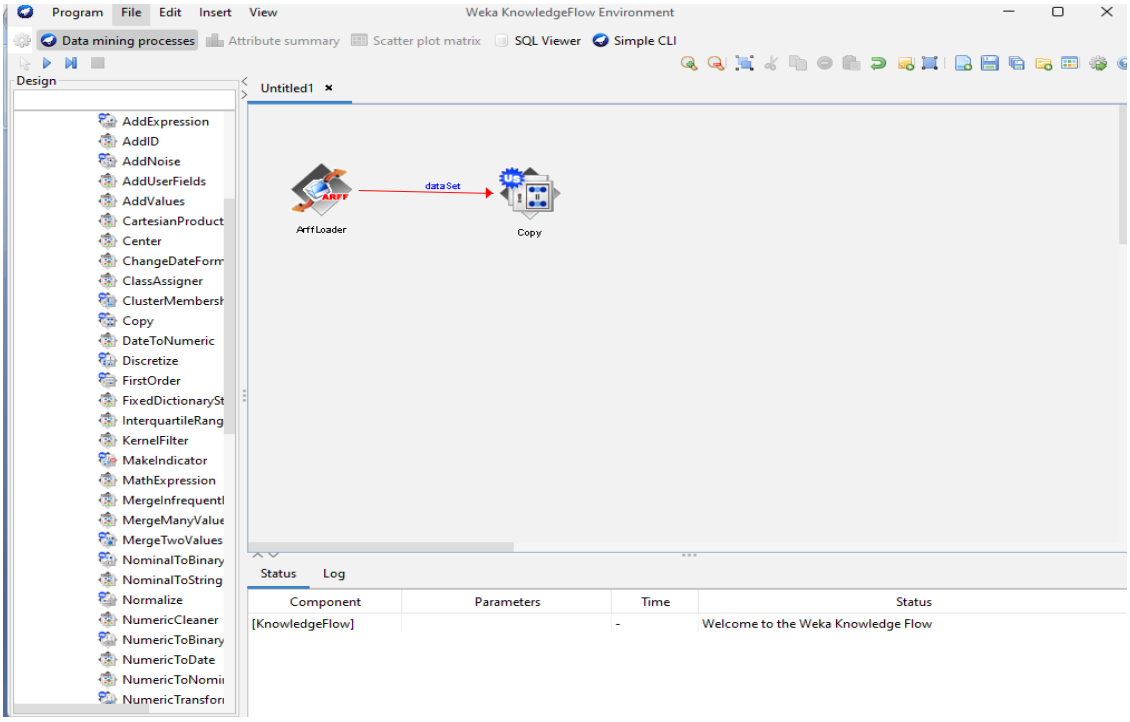


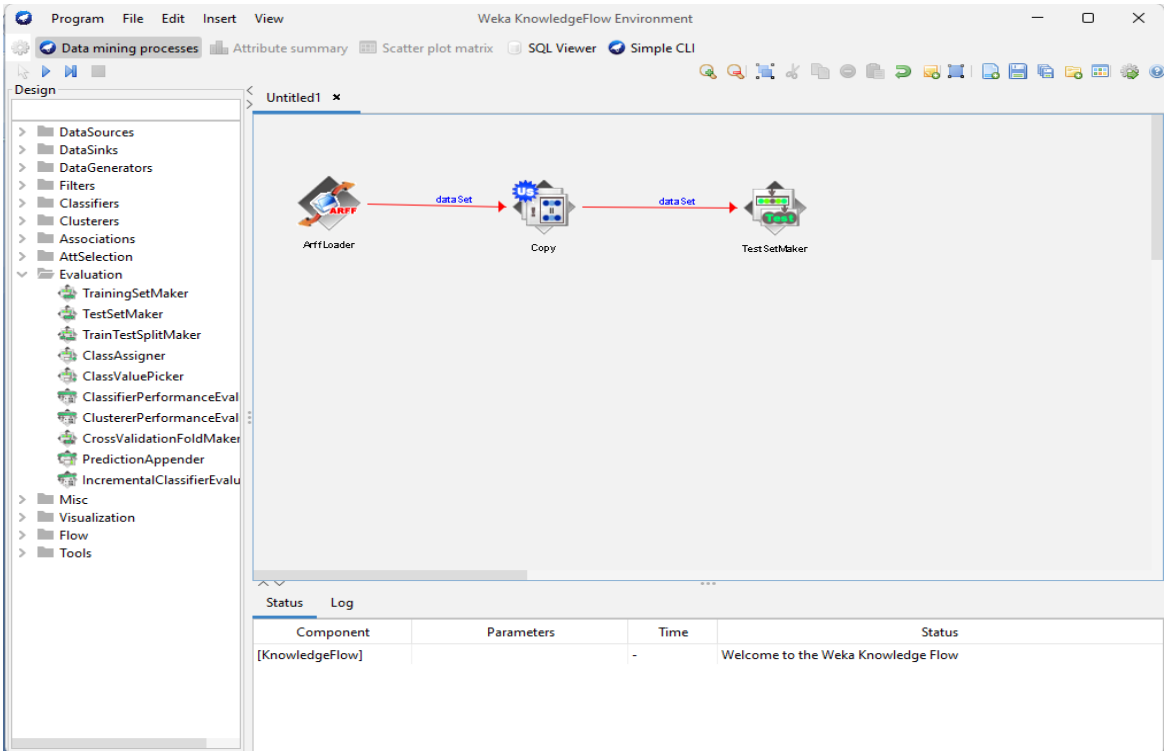
➔ *To Copy Attribute:*

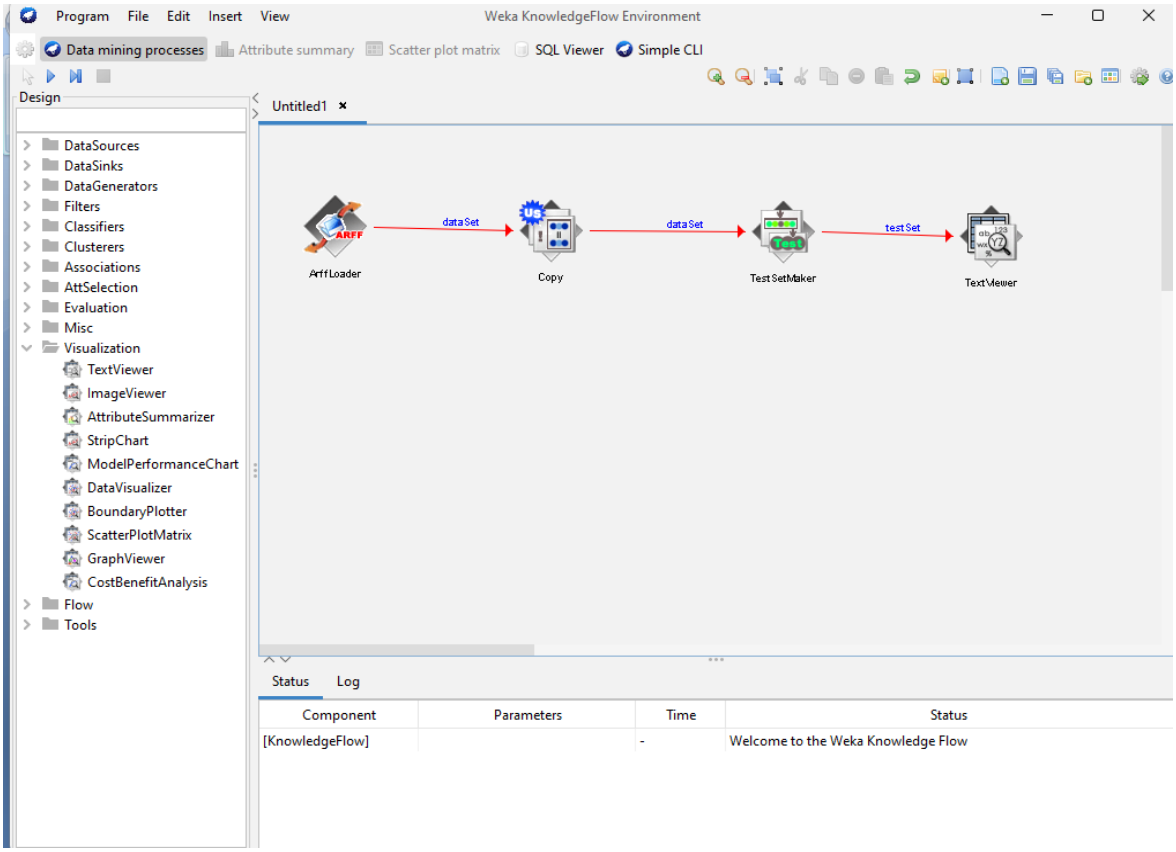


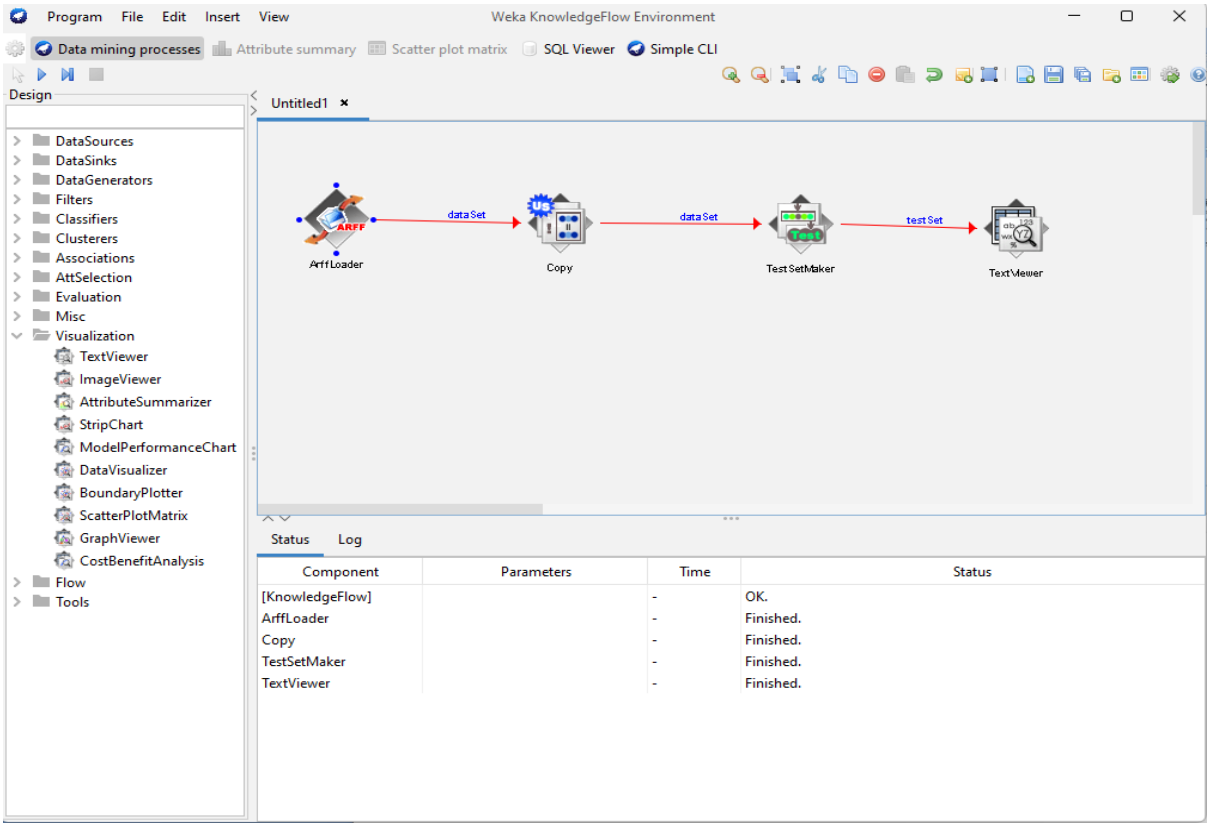

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
Filters
AllFilter
MultiFilter
RenameRelation
supervised
unsupervised
attribute
Add
AddCluster
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
Untitled1
ArffLoader
ArffLoader options
About
Reads a source that is in arff (attribute relation file format) format.
More
retainStringVals
False
useRelativePath
False
Filename
C:\Program Files\Weka-3-8-6\data\iris.arff
Browse...
OK
Cancel
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

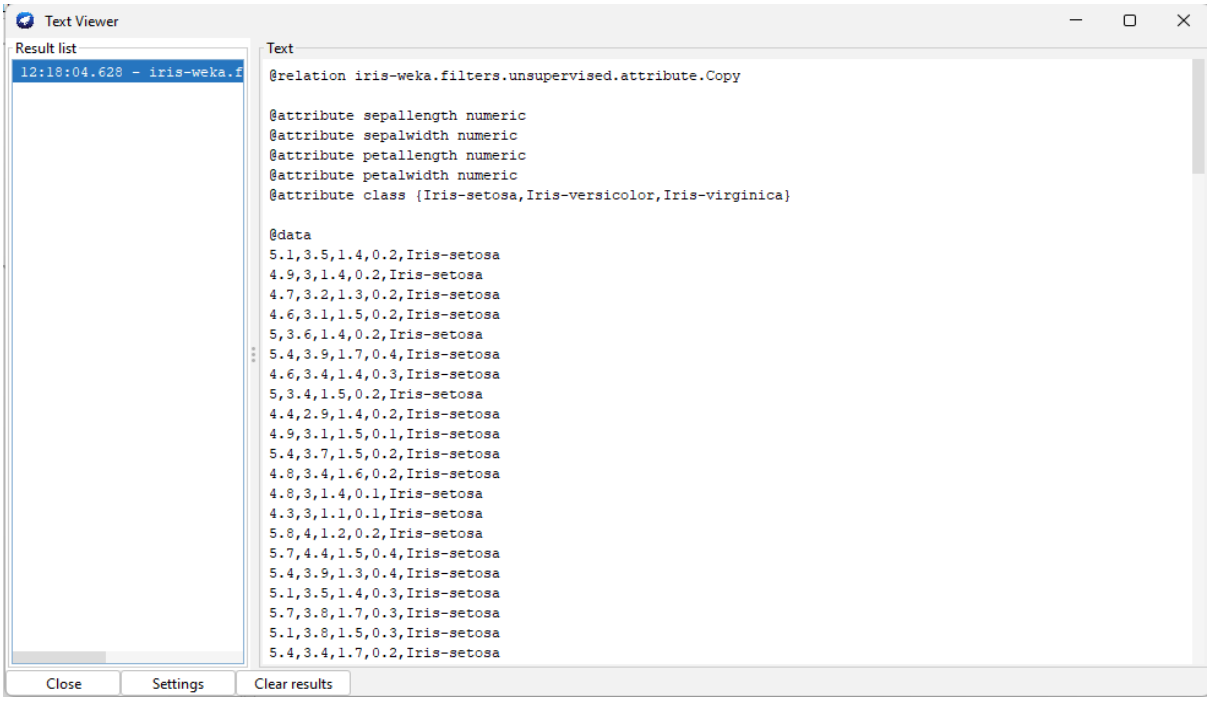

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
MergeTwoValues
NominalToBinary
NominalToString
Normalize
NumericCleaner
NumericToBinary
NumericToDate
NumericToNomi
NumericTransfor
Untitled1
ArffLoader
Copy
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
AddExpression
AddID
AddNoise
AddUserFields
AddValues
CartesianProduct
Center
ChangeDateForm
ClassAssigner
ClusterMembersh
Copy
DateToNumeric
Discretize
FirstOrder
FixedDictionarySt
InterquartileRang
KernelFilter
MakeIndicator
MathExpression
MergeInfrequentl
MergeManyValue
MergeTwoValues
NominalToBinary
NominalToString
Normalize
NumericCleaner
NumericToBinary
NumericToDate
NumericToNomi
NumericTransfor
Untitled1
ArffLoader
data Set
Copy
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
TrainingSetMaker
TestSetMaker
TrainTestSplitMaker
ClassAssigner
ClassValuePicker
ClassifierPerformanceEval
ClustererPerformanceEval
CrossValidationFoldMaker
PredictionAppender
IncrementalClassifierEvalu
Misc
Visualization
Flow
Tools
Untitled1
ArffLoader
data Set
Copy
data Set
TestSetMaker
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow


Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
Misc
Visualization
TextViewer
ImageViewer
AttributeSummarizer
StripChart
ModelPerformanceChart
DataVisualizer
BoundaryPlotter
ScatterPlotMatrix
GraphViewer
CostBenefitAnalysis
Flow
Tools
Untitled1
ArffLoader
data Set
Copy
data Set
TestSetMaker
test Set
TextViewer
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
Welcome to the Weka Knowledge Flow

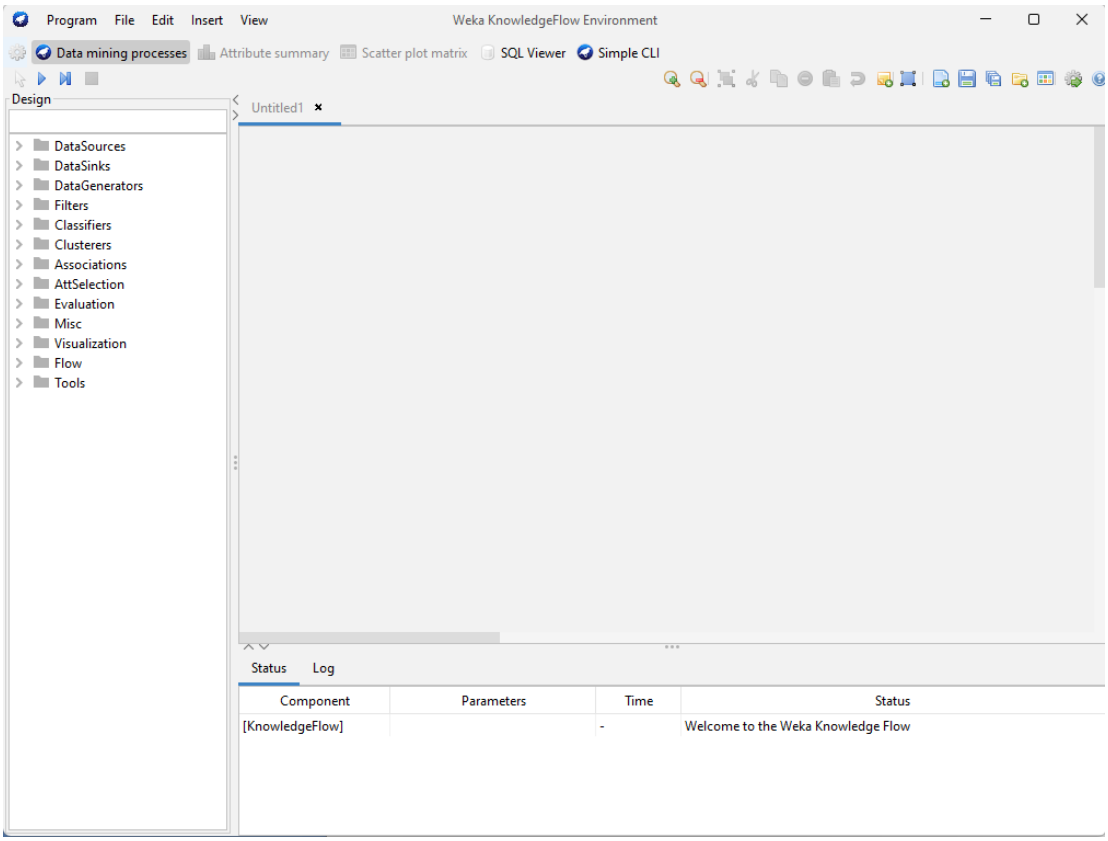

Program File Edit Insert View
Weka KnowledgeFlow Environment
Data mining processes
Attribute summary
Scatter plot matrix
SQL Viewer
Simple CLI
Design
DataSources
DataSinks
DataGenerators
Filters
Classifiers
Clusterers
Associations
AttSelection
Evaluation
Misc
Visualization
TextViewer
ImageViewer
AttributeSummarizer
StripChart
ModelPerformanceChart
DataVisualizer
BoundaryPlotter
ScatterPlotMatrix
GraphViewer
CostBenefitAnalysis
Flow
Tools
Untitled1
ArffLoader
data Set
Copy
data Set
TestSetMaker
test Set
TextViewer
Status
Log
Component
Parameters
Time
Status
[KnowledgeFlow]
-
OK.
ArffLoader
-
Finished.
Copy
-
Finished.
TestSetMaker
-
Finished.
TextViewer
-
Finished.



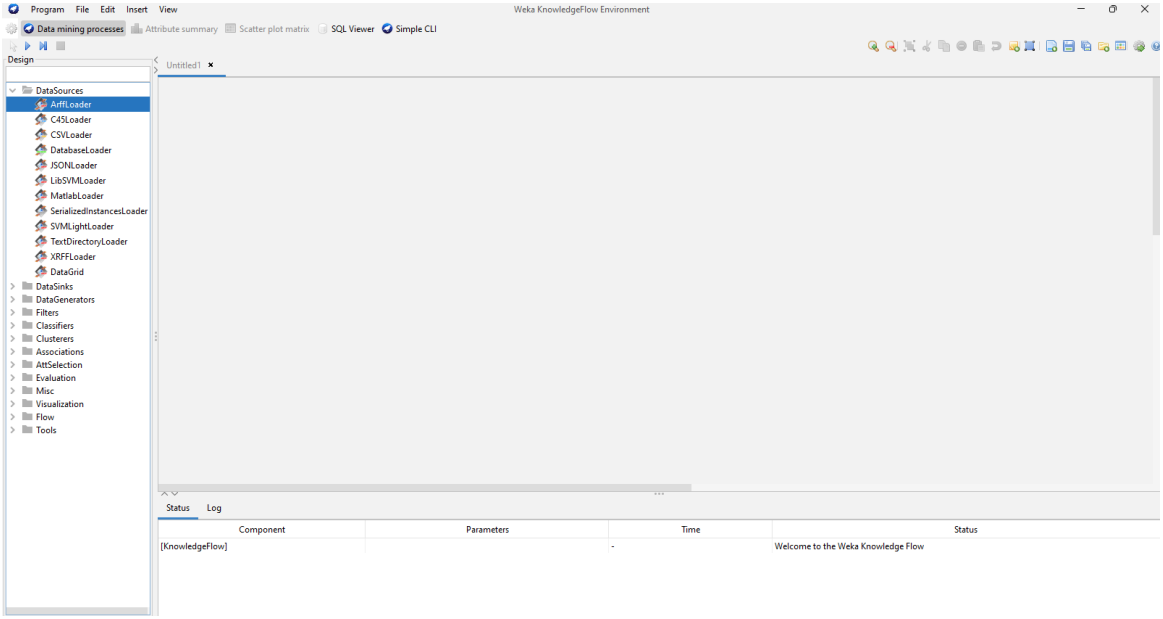
→*To Add Attribute*

1. 1. Open WEKA Tool and navigate to KnowledgeFlow.

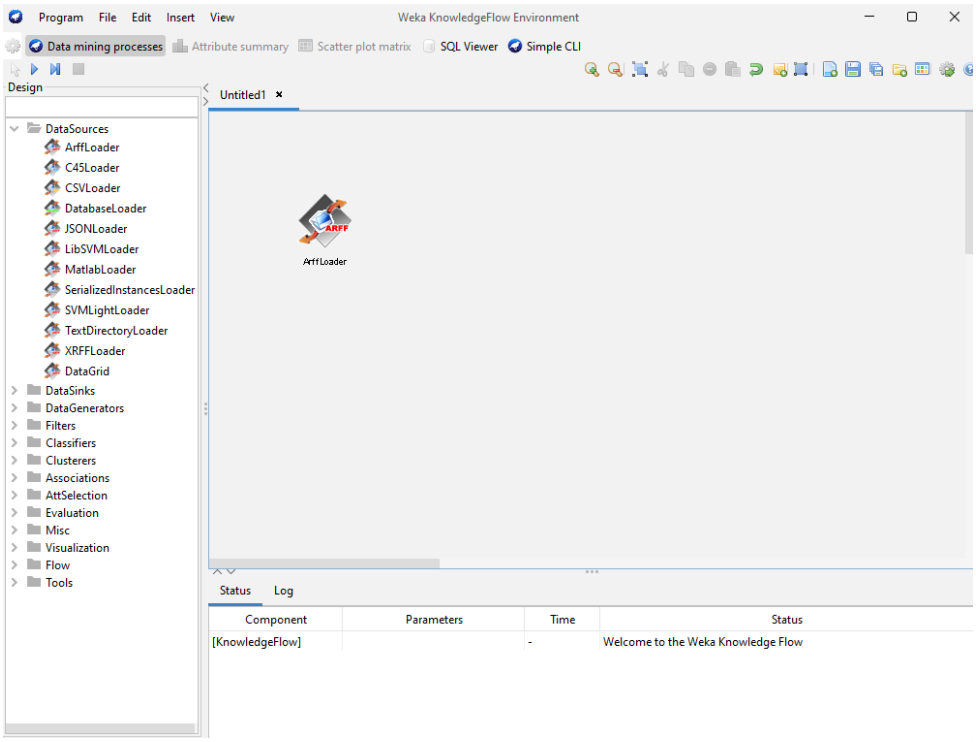
2. Knowledge Flow Environment views :,



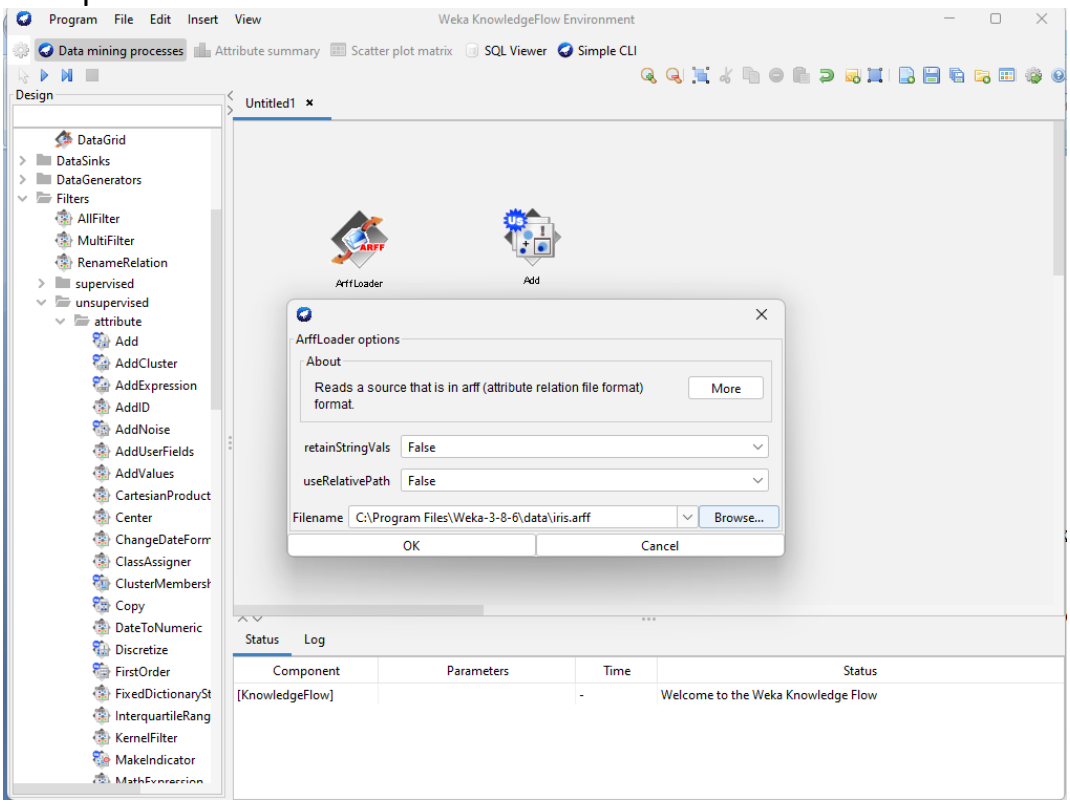
3. . To create a model's Knowledge Flow, pick DataSources in the Design box. Expand data sources. To pick the dataset, use either ArffLoader or CSV Loader.



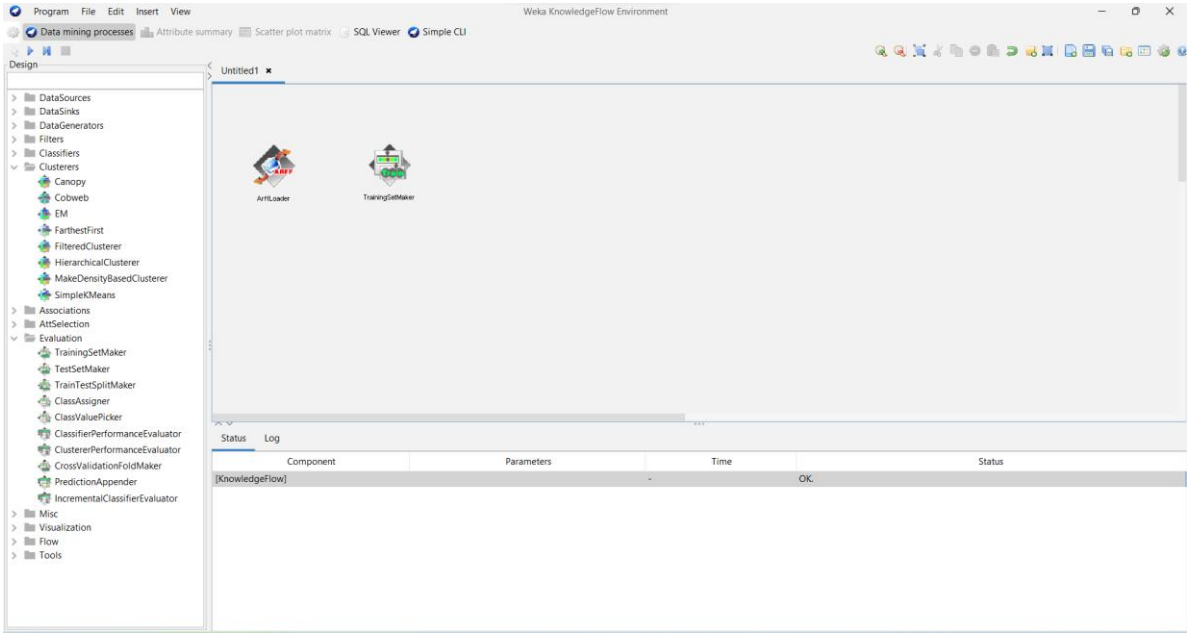
4. Click on the canvas window to add the Loader to the Knowledge Flow.



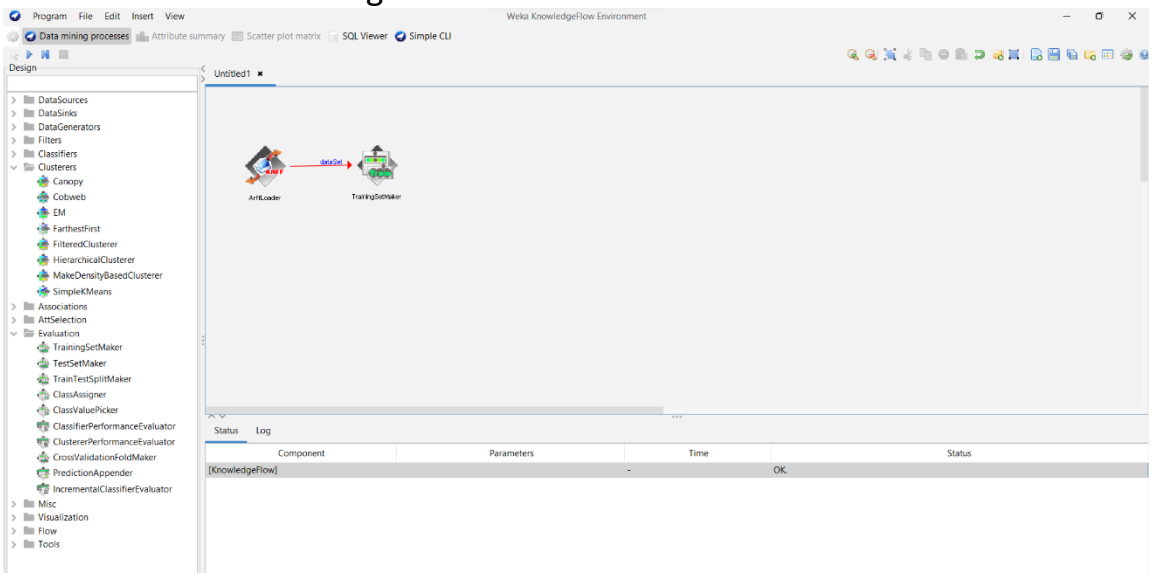
5. To load the dataset in ArffLoader, right-click on it, select Configure, and then pick the dataset.



6. To evaluate the dataset, select Evaluation > TrainingSetMaker in the Design Window.

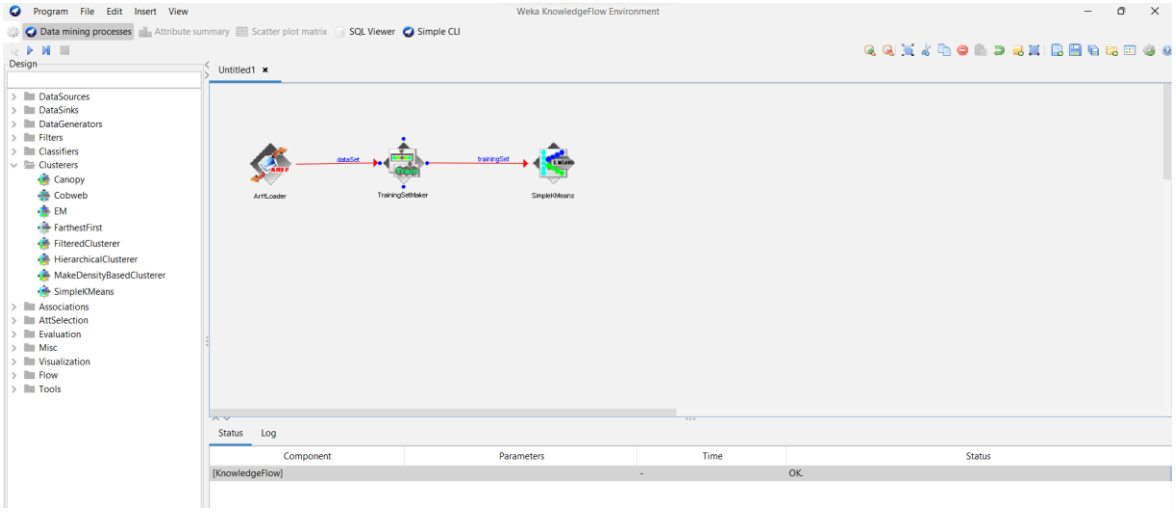


7. To link ArffLoader and TrainingSetMaker, right-click on Arffoader>dataSet and then choose TrainingSetMaker.

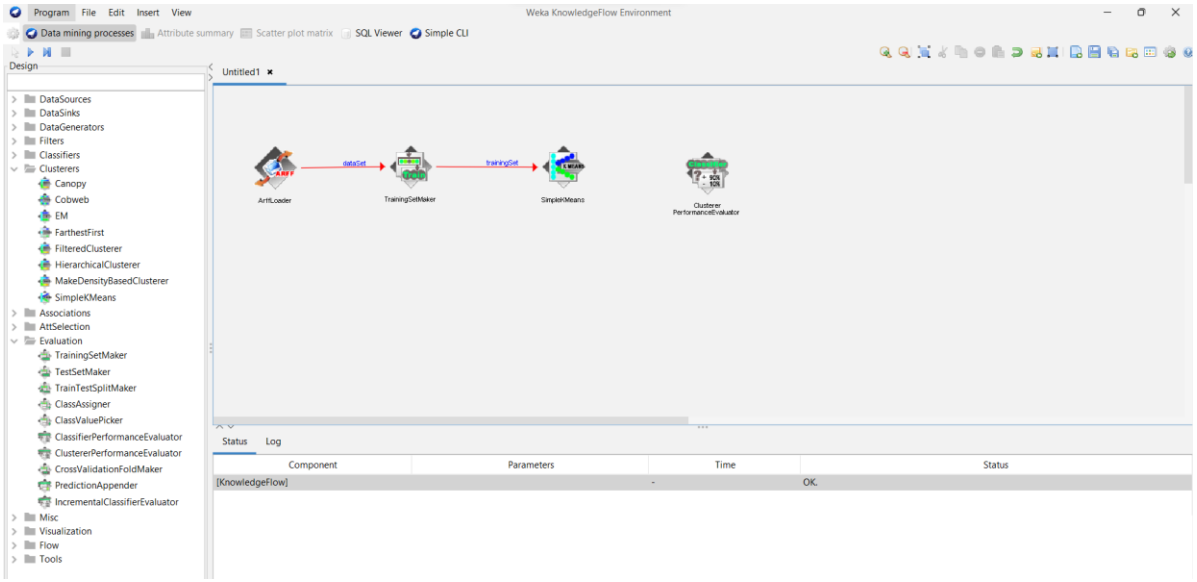


8. To add a cluster, navigate to Clusters>SimpleKMeans and set it on the Canvas Window.

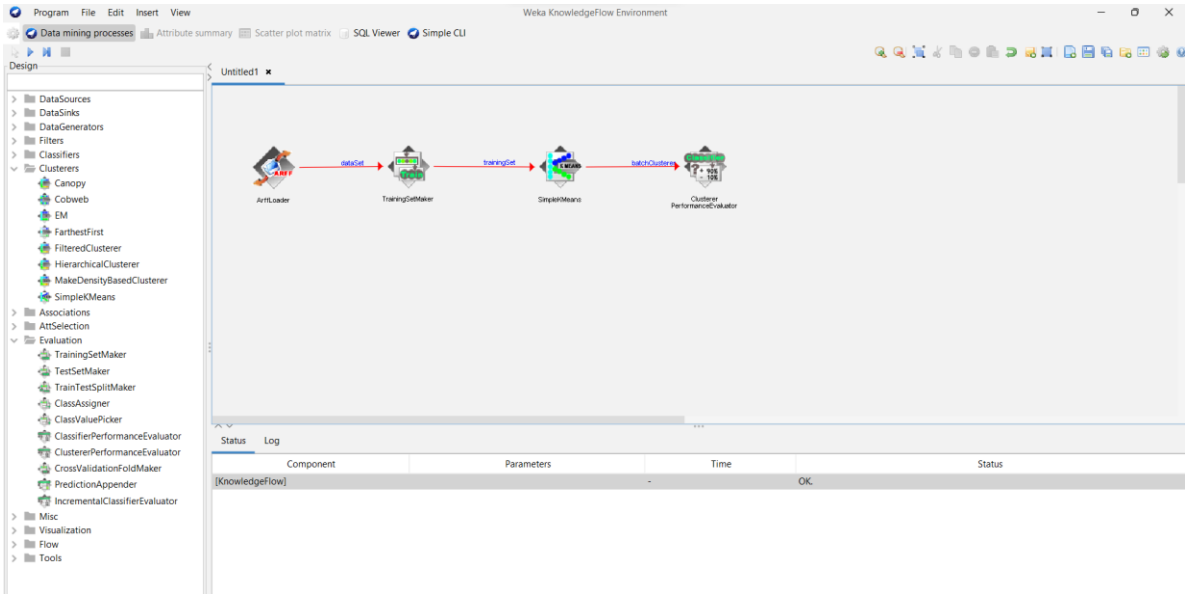
9. To link SimpleKMeans and TrainingSetMaker, right-click on TrainingSetMaker >trainingSet and then select TrainingSetMaker.



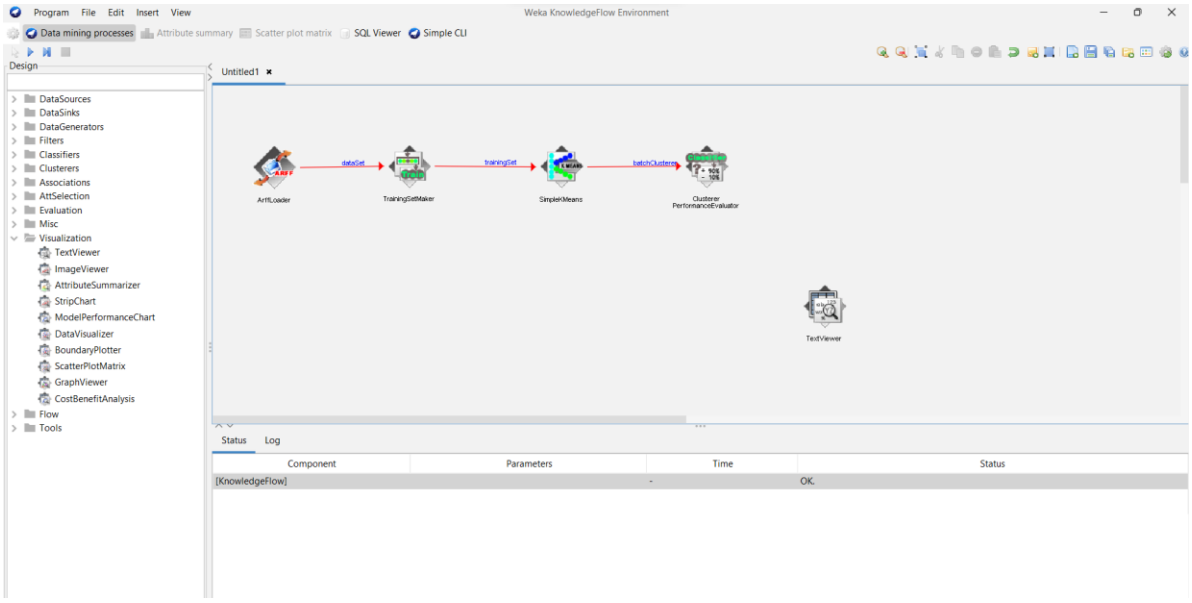
10. To assess the Cluster's performance, add a ClusterPerformanceEvaluator from the Design Window, choose Evaluation, and place it on Canvas.



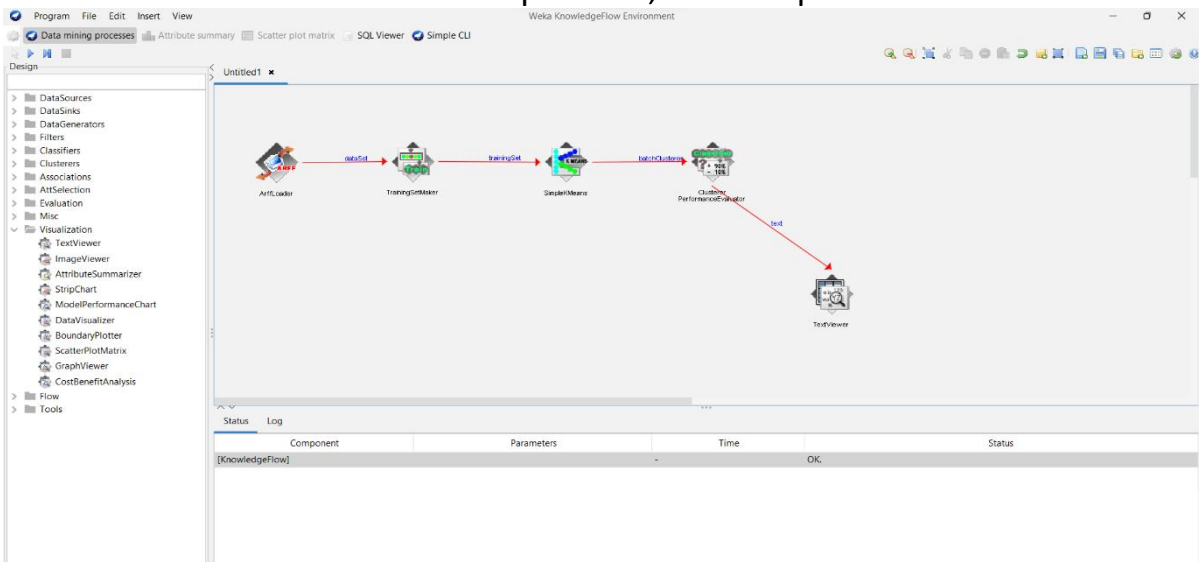
11. To connect Simple K Means and ClusterPerformanceEvaluator, right-click on SimpleKMeans and choose batchCluster, then link it to ClusterPerformanceEvaluator.



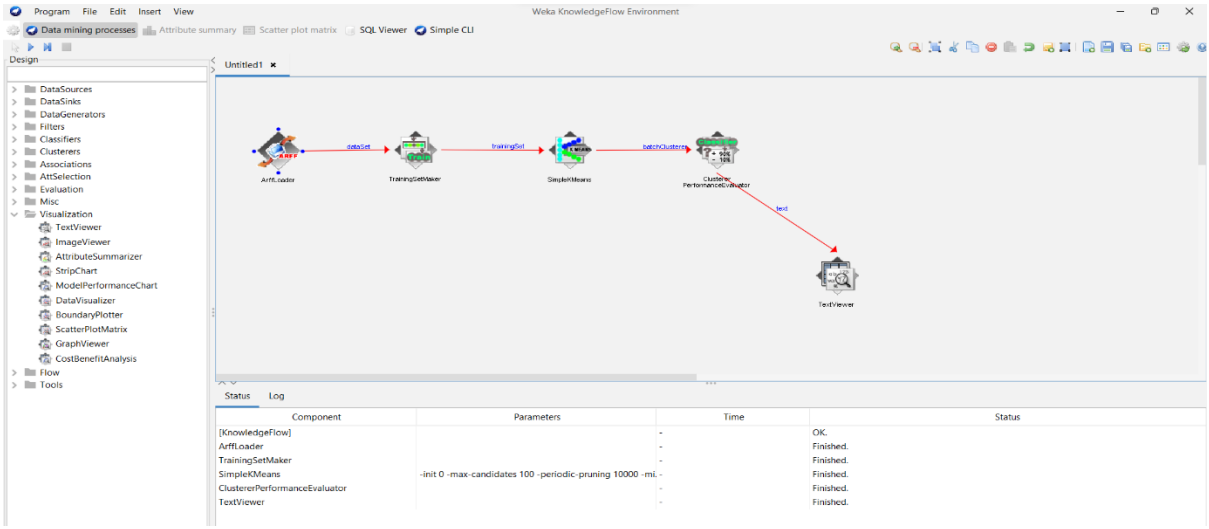
12. To view the results, we need to use Visualization. Select Visualization>TextViewer and place it on the canvas window.



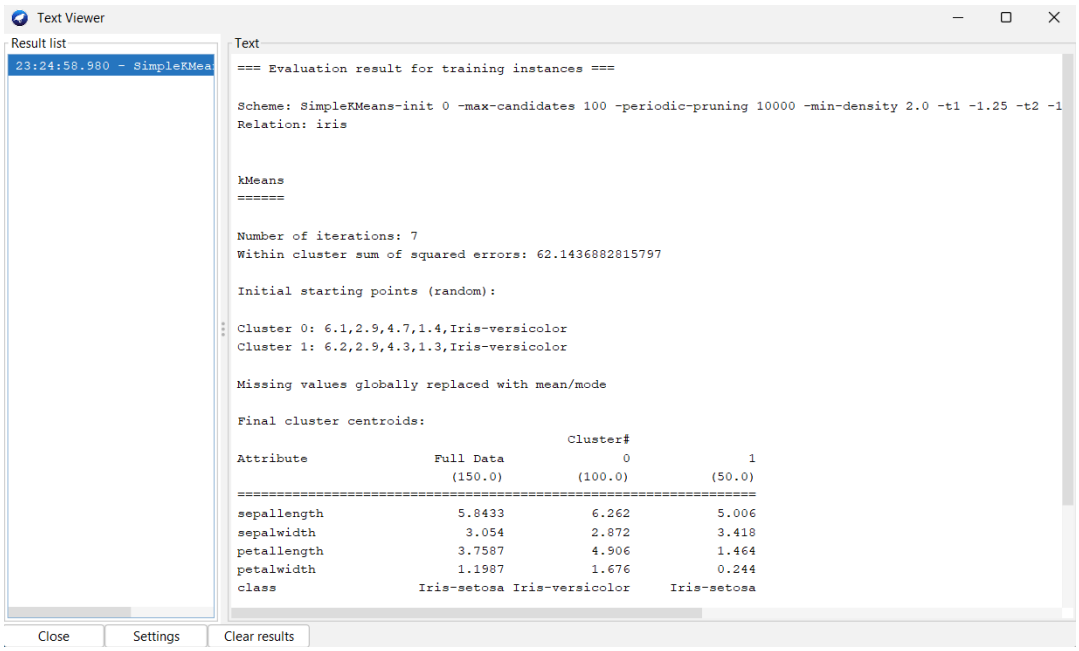
13. To connect ClusterPerformanceEvaluator and TextViewer, right-click on ClusterPerformanceEvaluator and pick text, then drop it onto TextViewer.

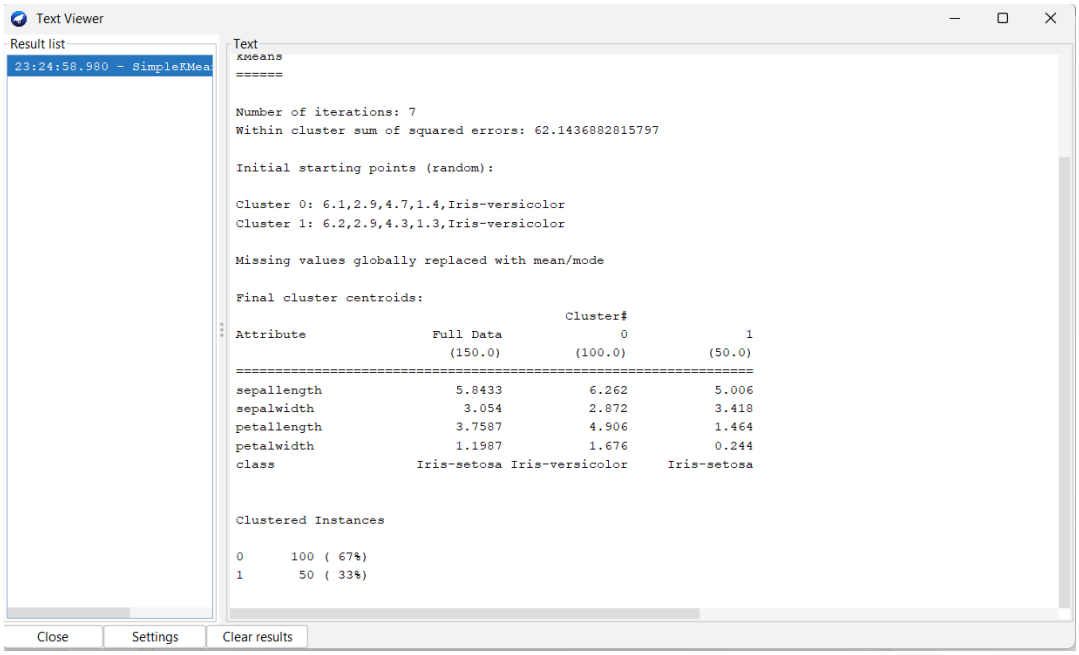


14. In the beginning the Knowledge Flow, click the Play button in the Menu bar.

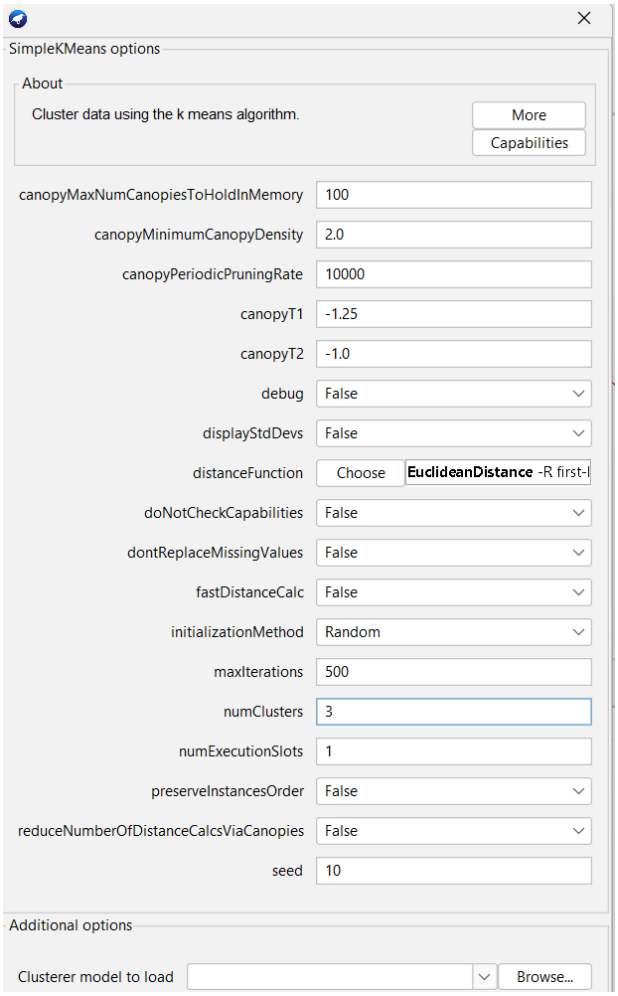


15. To examine the results, right-click on TextViewer and select Show Results.

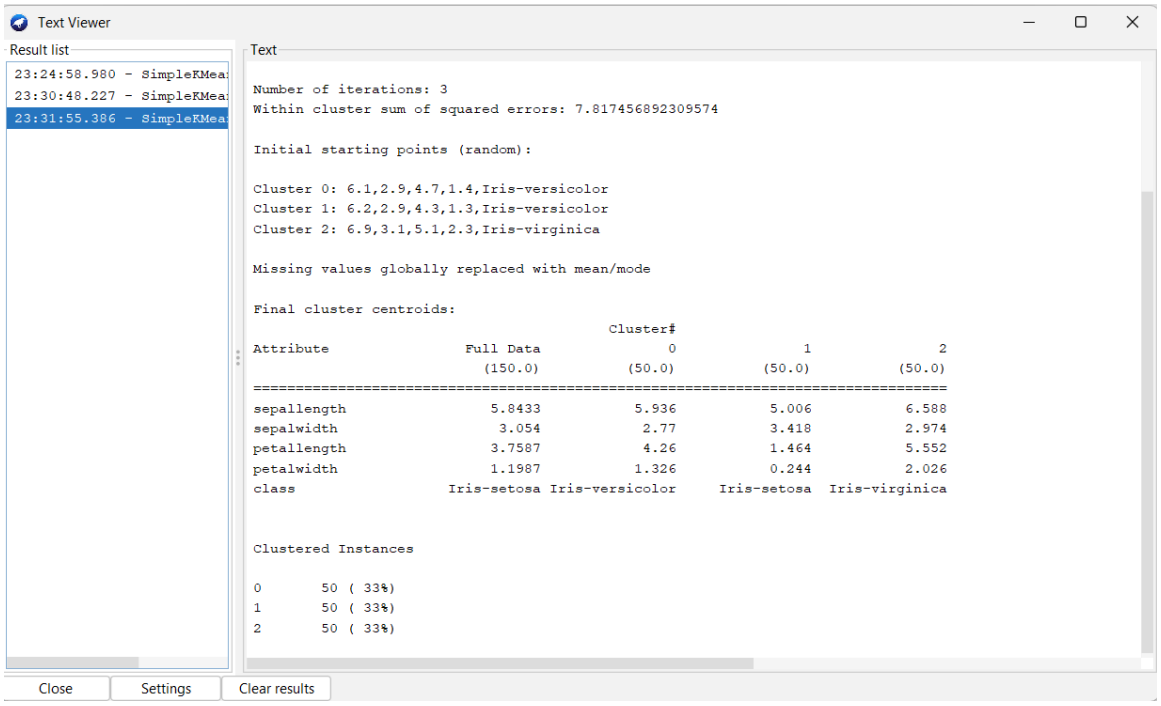




15. We can configure SimpleKMeans to obtain Clustered Instances, right click on SimpleKMeans modify the number at numClusters



16. To view findings, re-run the KnowledgeFlow and right-click on the TextViewer.



The numClusters changed to 3. So these are the Clustered Instances findings.