Index

Practical 1

Practical of Data Exploration (Statistical Analysis) and Data Pre-processing (Data Transformation and Dimension Reduction)

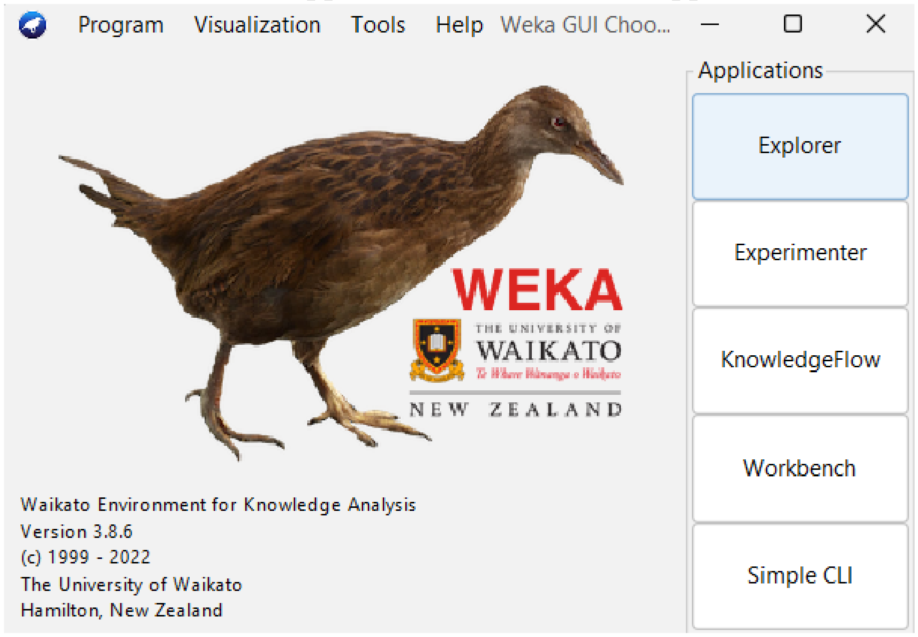
```
 import numpy as np
from scipy import stats

data = [34, 45, 32, 48, 22, 55, 36, 38, 40, 28, 60]

mean = np.mean(data)
median = np.median(data)
std_dev = np.std(data)
variance = np.var(data)
min_value = min(data)
max_value = max(data)
range_value = max_value - min_value

t_statistic, p_value = stats.ttest_1samp(data, popmean=40)

print("Data : ", data)
print("Mean : ", mean)
print("Median : ", median)
print("Standard deviation : ", std_dev)
print("Variance : ", variance)
print("Minimum value : ", min_value)
print("Maximum value : ", max_value)
print("Range : ", range_value)
print("T-Statistic : ", t_statistic)
print("P-Value : ", p_value)

#Perform a normality test
shapiro_stat, shapiro_p = stats.shapiro(data)
if shapiro_p > 0.05 :
   print("Data is normally distributed (Shapiro-Wilk testp -value = ", shapiro_p, ")")
else :
   print("Data is not normally distributed (Shapiro-Wilk testp -value = ", shapiro_p, ")")
```

```
Data :  [34, 45, 32, 48, 22, 55, 36, 38, 40, 28, 60]
Mean :  39.81818181818182
Median :  38.0
Standard deviation :  10.877985958457415
Variance :  118.33057851239668
Minimum value :  22
Maximum value :  60
Range :  38
T-Statistic :  -0.05285533340195602
P-Value :  0.9588881490809511
Data is normally distributed (Shapiro-Wilk testp -value =  0.9621524810791016 )
```

Practical 3
Implementation of pre-processing in WEKA.

Step 1 :

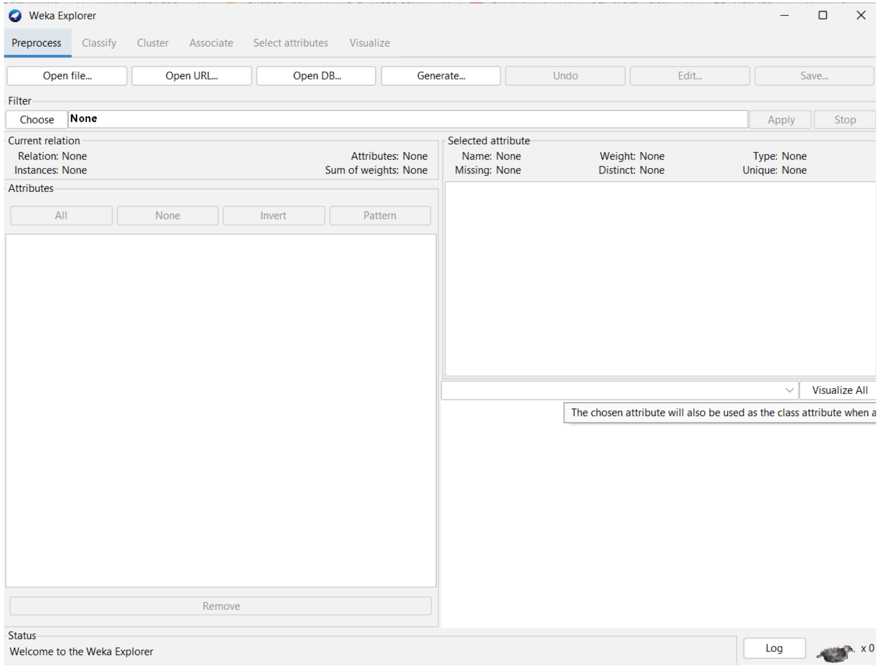
Open the WEKA application and the first page is displayed where you have the option to choose from various applications that WEKA supports.



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Step 2 :

Click on "Explorer" and the below image is shown



Filter: None

Relation: None   Attributes: None
Instances: None   Sum of weights: None

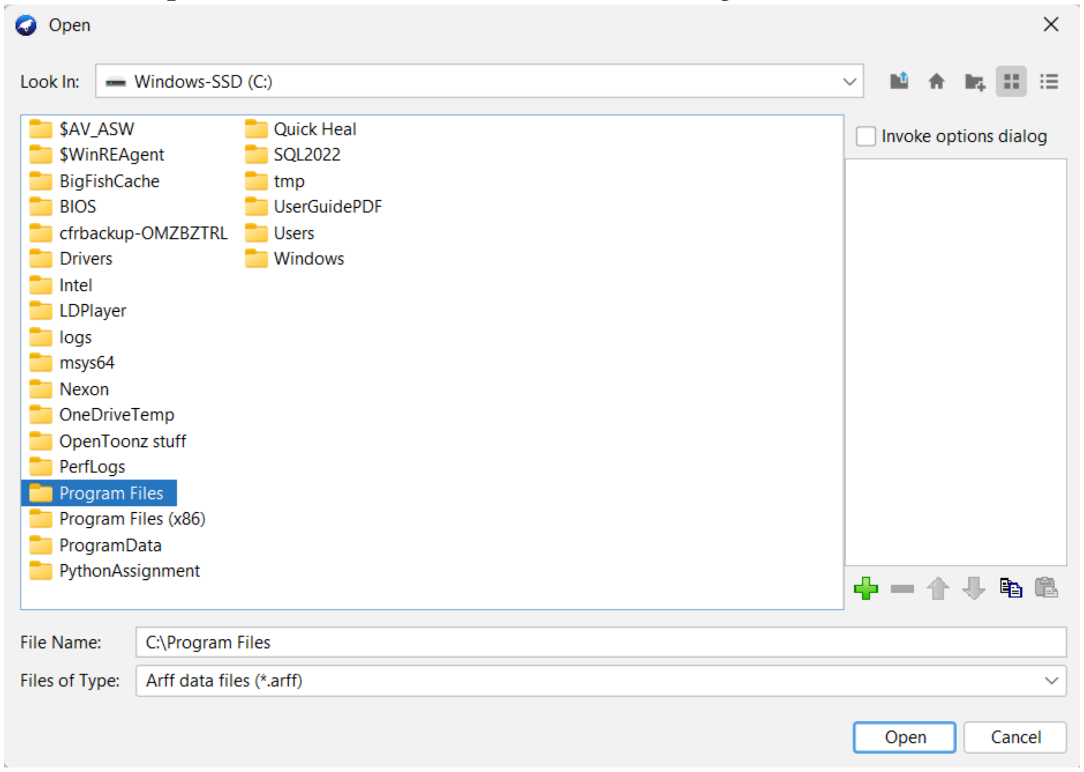
Name: None   Weight: None   Type: None
Missing: None   Distinct: None   Unique: None

The chosen attribute will also be used as the class attribute when a

Welcome to the Weka Explorer

Step 3 :

Click on "Open File" and then double click on "Program Files"



Open

Look In: Windows-SSD (C:)

$AV_ASW
$WinREAgent
BigFishCache
BIOS
cfrbackup-OMZBZTRL
Drivers
Intel
LDPlayer
logs
msys64
Nexon
OneDriveTemp
OpenToonz stuff
PerfLogs
Program Files
Program Files (x86)
ProgramData
PythonAssignment
Quick Heal
SQL2022
tmp
UserGuidePDF
Users
Windows

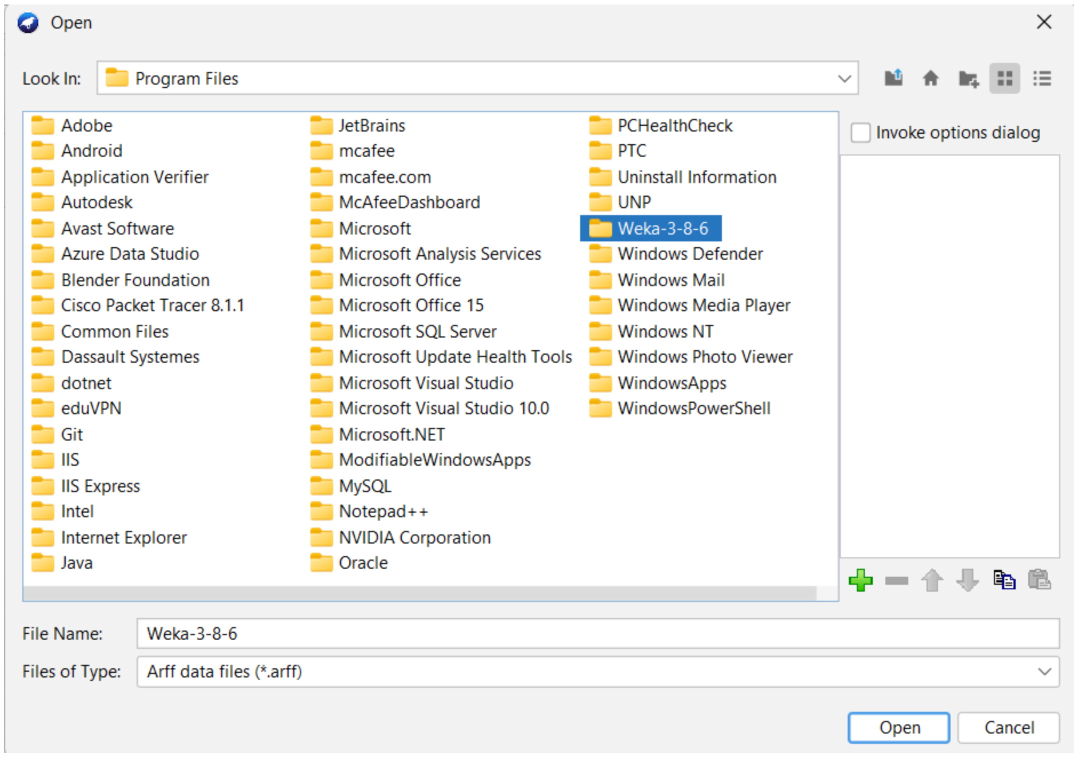
Invoke options dialog

File Name: C:\Program Files

Files of Type: Arff data files (*.arff)

Open Cancel

Then double click on "Weka-3-8-6"



Open

Look In: Program Files

Adobe
Android
Application Verifier
Autodesk
Avast Software
Azure Data Studio
Blender Foundation
Cisco Packet Tracer 8.1.1
Common Files
Dassault Systemes
dotnet
eduVPN
Git
IIS
IIS Express
Intel
Internet Explorer
Java
JetBrains
mcafee
mcafee.com
McAfeeDashboard
Microsoft
Microsoft Analysis Services
Microsoft Office
Microsoft Office 15
Microsoft SQL Server
Microsoft Update Health Tools
Microsoft Visual Studio
Microsoft Visual Studio 10.0
Microsoft.NET
ModifiableWindowsApps
MySQL
Notepad++
NVIDIA Corporation
Oracle
PCHealthCheck
PTC
Uninstall Information
UNP
Weka-3-8-6
Windows Defender
Windows Mail
Windows Media Player
Windows NT
Windows Photo Viewer
WindowsApps
WindowsPowerShell

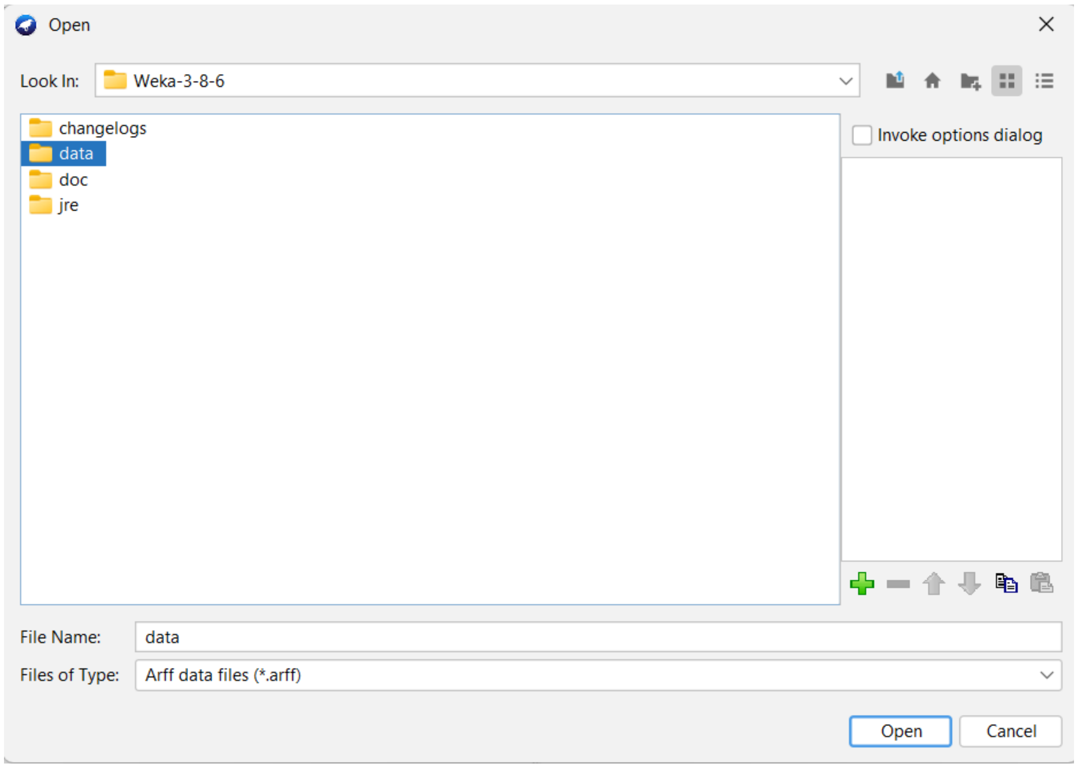
Invoke options dialog

File Name: Weka-3-8-6

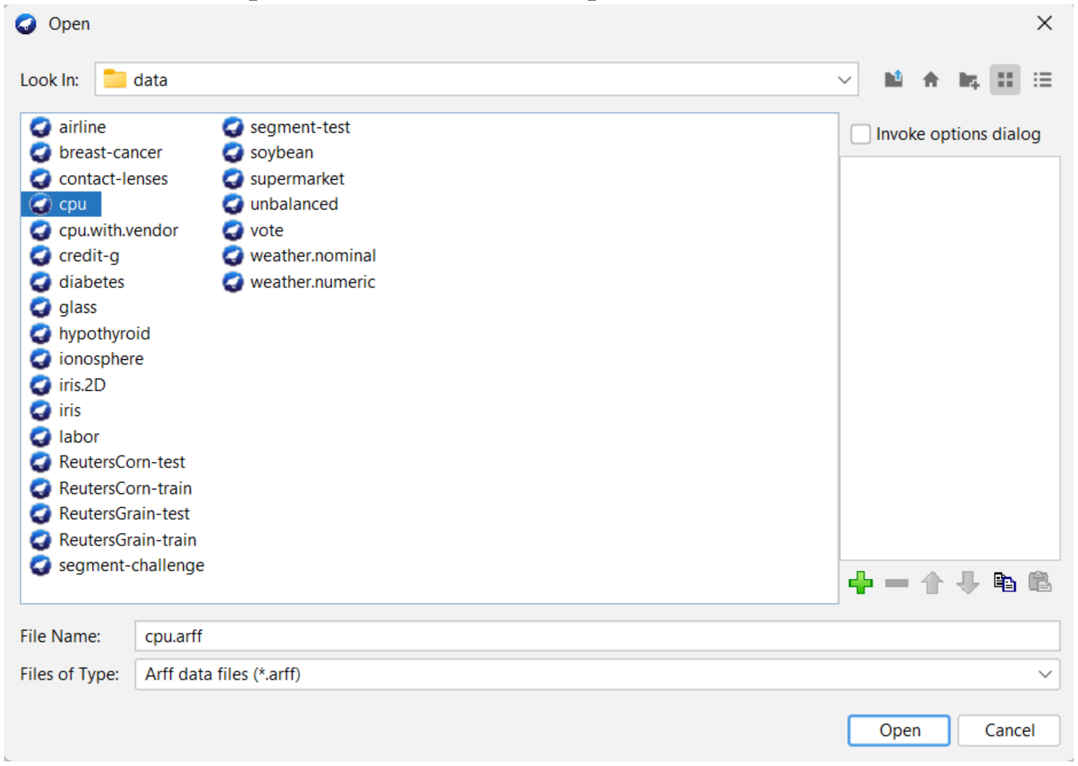
Files of Type: Arff data files (*.arff)

Open Cancel

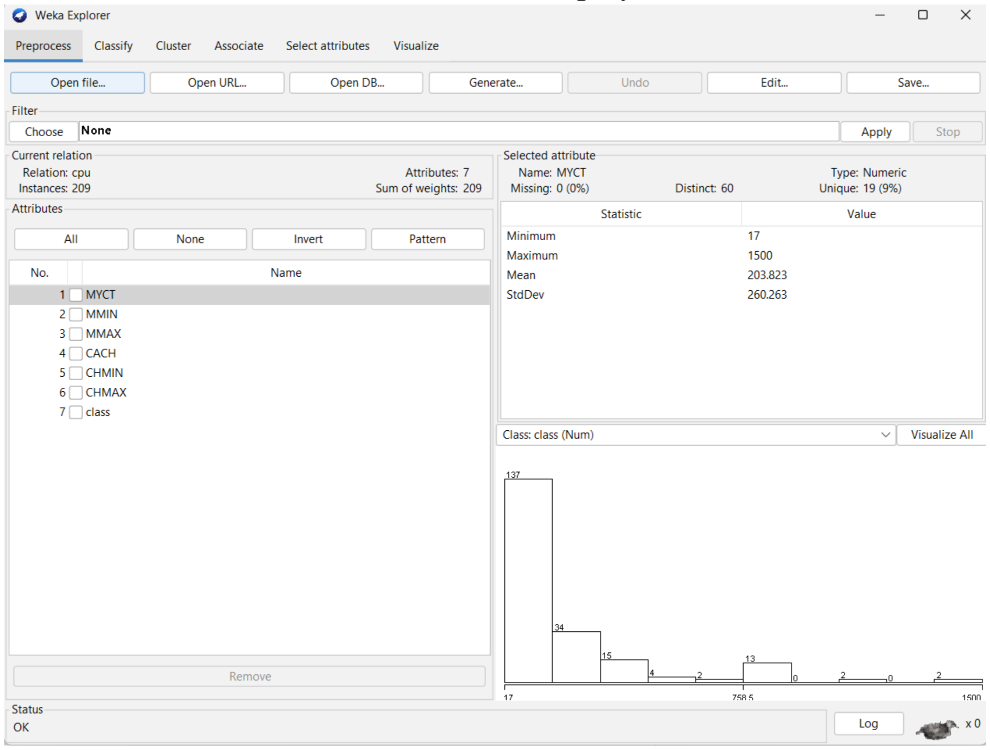
Then double click on “data” folder



Then select the “cpu” dataset and click “Open”



The dataset is added and this screen is displayed



Filter: None

Current relation
Relation: cpu
Instances: 209
Attributes: 7
Sum of weights: 209

Selected attribute
Name: MYCT
Missing: 0 (0%)
Distinct: 60
Type: Numeric
Unique: 19 (9%)

Statistic	Value
Minimum	17
Maximum	1500
Mean	203.823
StdDev	260.263

Attributes

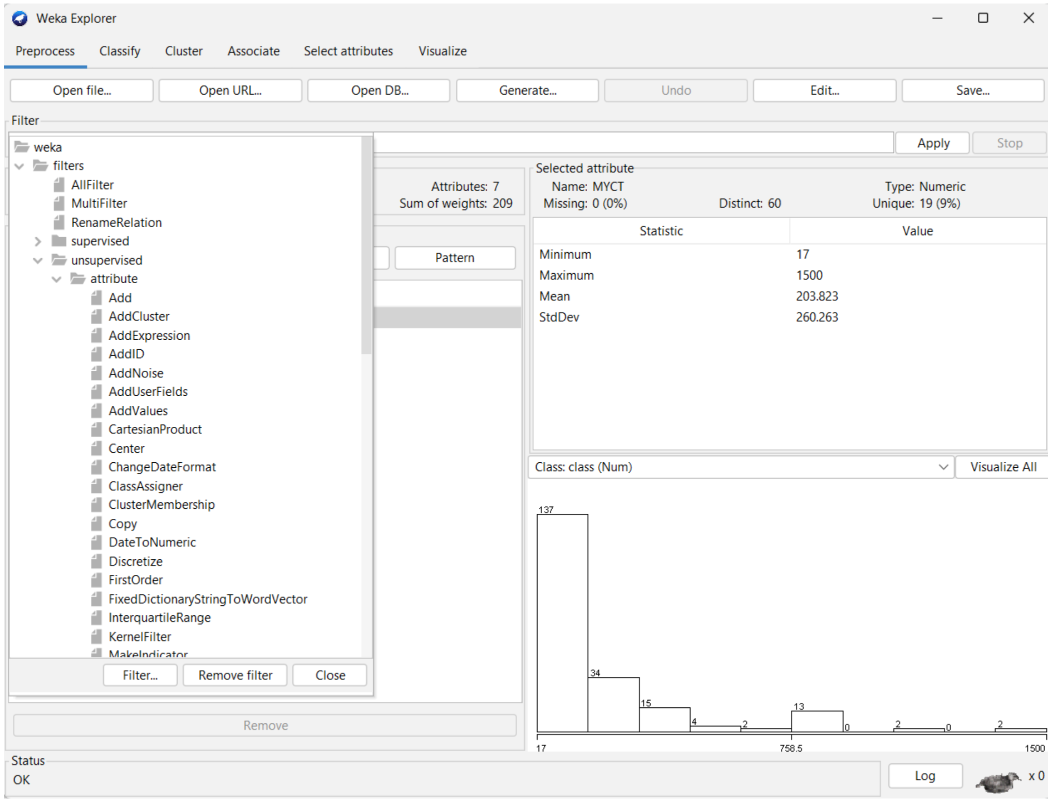
No.	Name
1	MYCT
2	MMIN
3	MMAX
4	CACH
5	CHMIN
6	CHMAX
7	class

Class: class (Num)

Status
OK

Step 4 :

Click on the "Choose" button. Choose the "Normalize" filter and then click "Apply"



Attributes: 7
Sum of weights: 209

Selected attribute
Name: MYCT
Missing: 0 (0%)
Distinct: 60
Type: Numeric
Unique: 19 (9%)

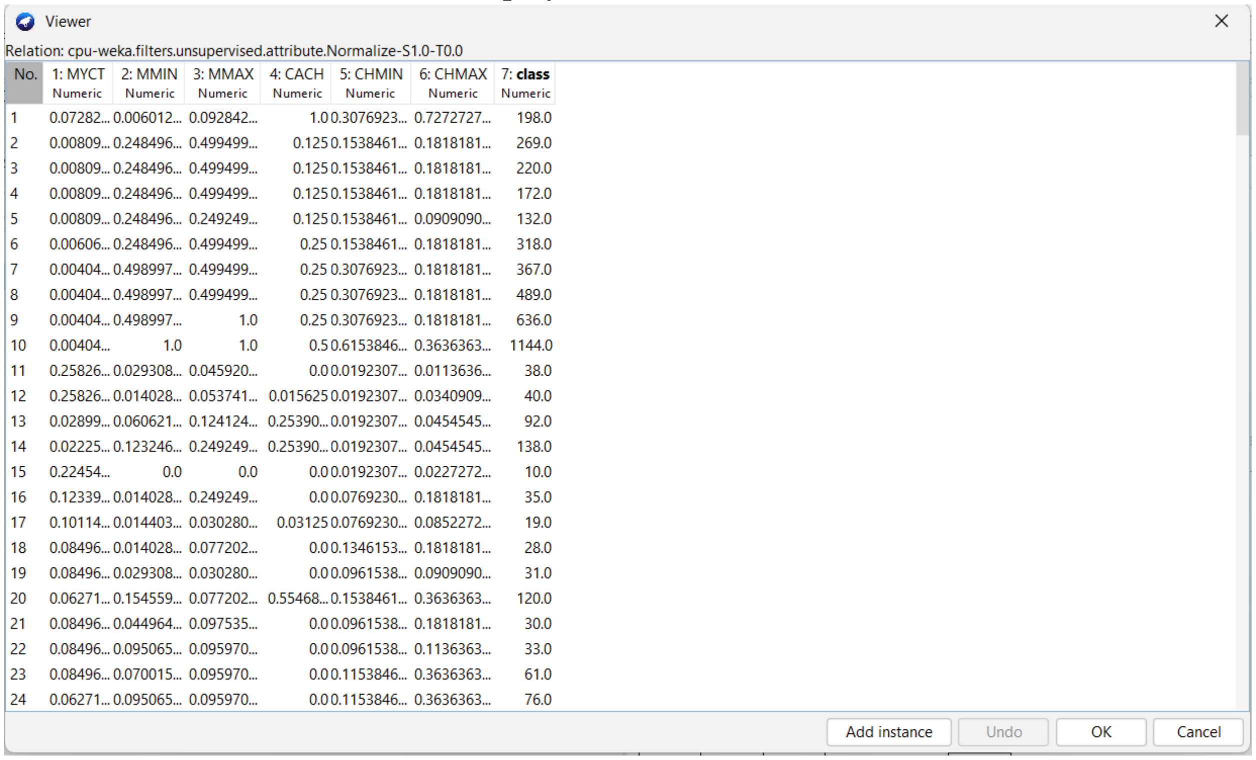
Statistic	Value
Minimum	17
Maximum	1500
Mean	203.823
StdDev	260.263

Class: class (Num)

Status
OK

Step 5 :

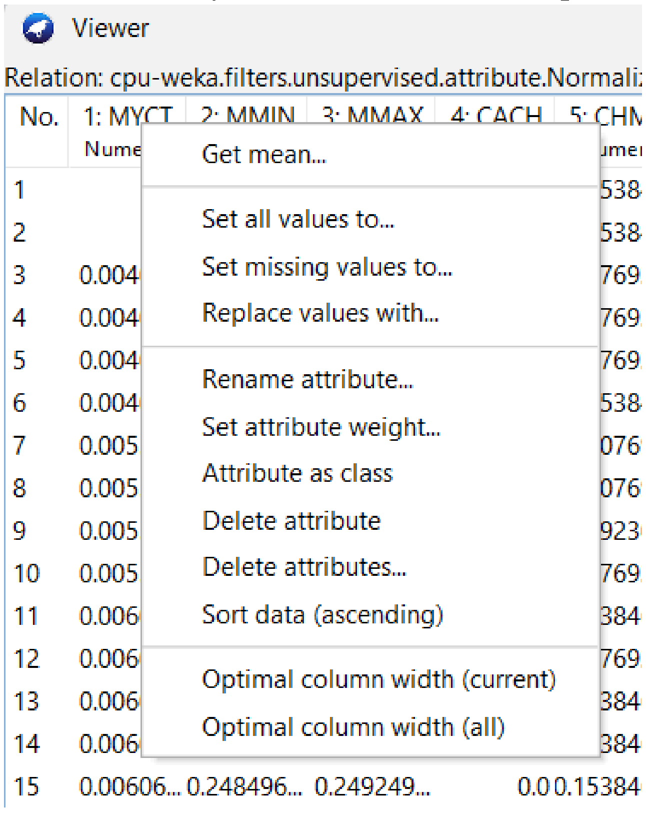
Click on “Edit” and the screen is displayed.



Relation: cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

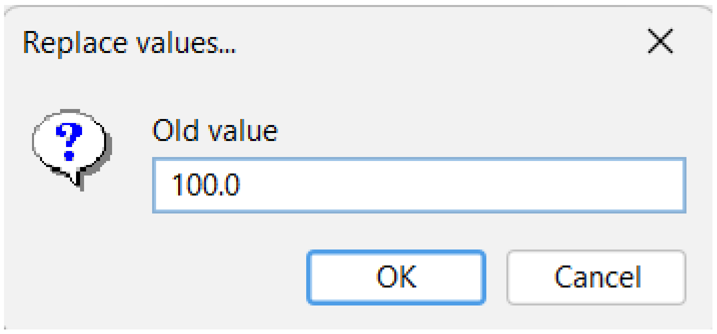
No.	1: MYCT Numeric	2: MMIN Numeric	3: MMAX Numeric	4: CACH Numeric	5: CHMIN Numeric	6: CHMAX Numeric	7: **class** Numeric
1	0.07282...	0.006012...	0.092842...	1.0	0.3076923...	0.7272727...	198.0
2	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.1818181...	269.0
3	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.1818181...	220.0
4	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.1818181...	172.0
5	0.00809...	0.248496...	0.249249...	0.125	0.1538461...	0.0909090...	132.0
6	0.00606...	0.248496...	0.499499...	0.25	0.1538461...	0.1818181...	318.0
7	0.00404...	0.498997...	0.499499...	0.25	0.3076923...	0.1818181...	367.0
8	0.00404...	0.498997...	0.499499...	0.25	0.3076923...	0.1818181...	489.0
9	0.00404...	0.498997...	1.0	0.25	0.3076923...	0.1818181...	636.0
10	0.00404...	1.0	1.0	0.5	0.6153846...	0.3636363...	1144.0
11	0.25826...	0.029308...	0.045920...	0.0	0.0192307...	0.0113636...	38.0
12	0.25826...	0.014028...	0.053741...	0.015625	0.0192307...	0.0340909...	40.0
13	0.02899...	0.060621...	0.124124...	0.25390...	0.0192307...	0.0454545...	92.0
14	0.02225...	0.123246...	0.249249...	0.25390...	0.0192307...	0.0454545...	138.0
15	0.22454...	0.0	0.0	0.0	0.0192307...	0.0227272...	10.0
16	0.12339...	0.014028...	0.249249...	0.0	0.0769230...	0.1818181...	35.0
17	0.10114...	0.014403...	0.030280...	0.03125	0.0769230...	0.0852272...	19.0
18	0.08496...	0.014028...	0.077202...	0.0	0.1346153...	0.1818181...	28.0
19	0.08496...	0.029308...	0.030280...	0.0	0.0961538...	0.0909090...	31.0
20	0.06271...	0.154559...	0.077202...	0.55468...	0.1538461...	0.3636363...	120.0
21	0.08496...	0.044964...	0.097535...	0.0	0.0961538...	0.1818181...	30.0
22	0.08496...	0.095065...	0.095970...	0.0	0.0961538...	0.1136363...	33.0
23	0.08496...	0.070015...	0.095970...	0.0	0.1153846...	0.3636363...	61.0
24	0.06271...	0.095065...	0.095970...	0.0	0.1153846...	0.3636363...	76.0

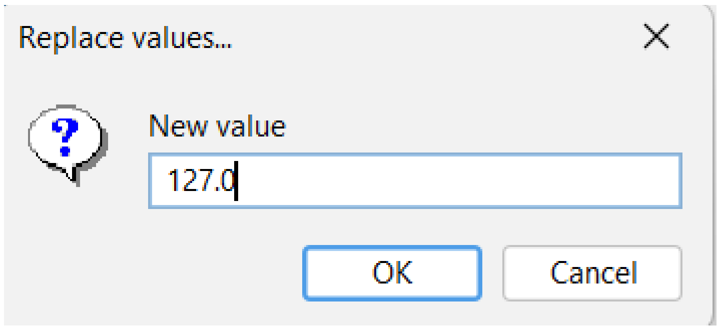
Here, Select any column and it can be replaced i.e select Replace values with



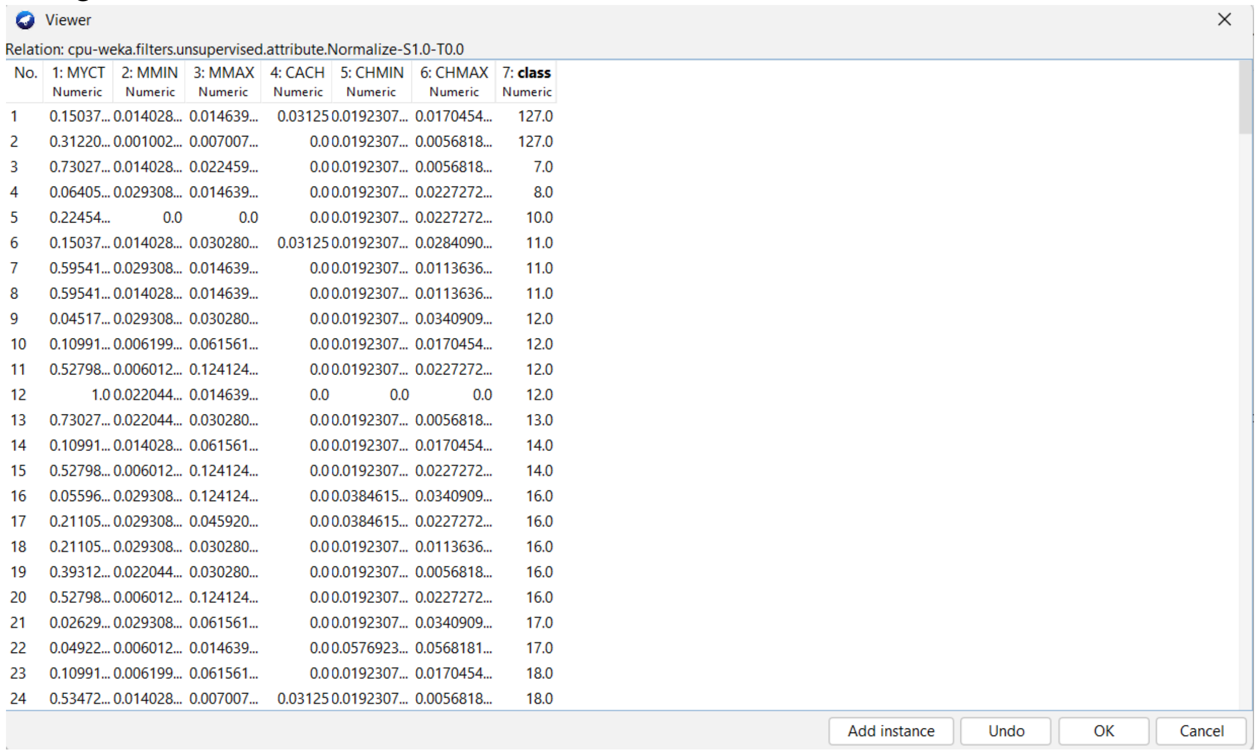
Relation: cpu-weka.filters.unsupervised.attribute.Normali

No.	1: MYCT Nume	2: MMIN	3: MMAX	4: CACH	5: CHN ume
1					538
2					538
3	0.004				769
4	0.004				769
5	0.004				769
6	0.004				538
7	0.005				076
8	0.005				076
9	0.005				923
10	0.005				769
11	0.006				384
12	0.006				769
13	0.006				384
14	0.006				384
15	0.00606...	0.248496...	0.249249...	0.0	0.15384





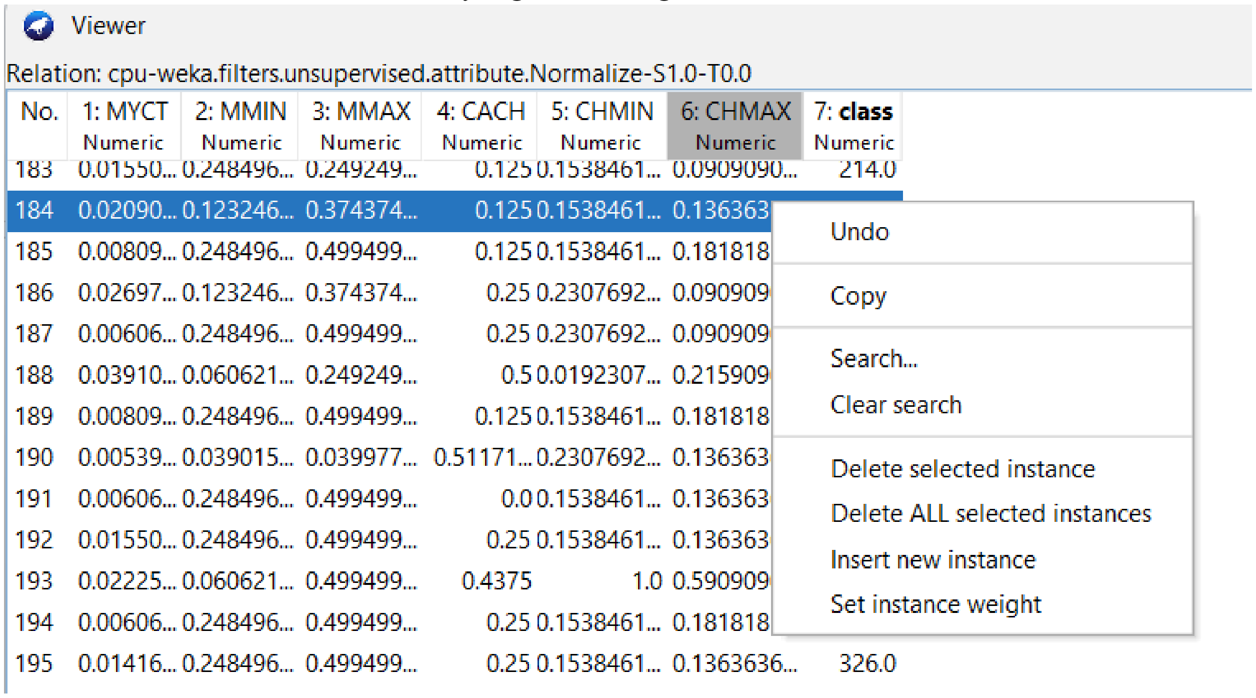
Changed values in the Class attribute



Relation: cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

No.	1: MYCT Numeric	2: MMIN Numeric	3: MMAX Numeric	4: CACH Numeric	5: CHMIN Numeric	6: CHMAX Numeric	7: **class** Numeric
1	0.15037...	0.014028...	0.014639...	0.03125	0.0192307...	0.0170454...	127.0
2	0.31220...	0.001002...	0.007007...	0.0	0.0192307...	0.0056818...	127.0
3	0.73027...	0.014028...	0.022459...	0.0	0.0192307...	0.0056818...	7.0
4	0.06405...	0.029308...	0.014639...	0.0	0.0192307...	0.0227272...	8.0
5	0.22454...	0.0	0.0	0.0	0.0192307...	0.0227272...	10.0
6	0.15037...	0.014028...	0.030280...	0.03125	0.0192307...	0.0284090...	11.0
7	0.59541...	0.029308...	0.014639...	0.0	0.0192307...	0.0113636...	11.0
8	0.59541...	0.014028...	0.014639...	0.0	0.0192307...	0.0113636...	11.0
9	0.04517...	0.029308...	0.030280...	0.0	0.0192307...	0.0340909...	12.0
10	0.10991...	0.006199...	0.061561...	0.0	0.0192307...	0.0170454...	12.0
11	0.52798...	0.006012...	0.124124...	0.0	0.0192307...	0.0227272...	12.0
12	1.0	0.022044...	0.014639...	0.0	0.0	0.0	12.0
13	0.73027...	0.022044...	0.030280...	0.0	0.0192307...	0.0056818...	13.0
14	0.10991...	0.014028...	0.061561...	0.0	0.0192307...	0.0170454...	14.0
15	0.52798...	0.006012...	0.124124...	0.0	0.0192307...	0.0227272...	14.0
16	0.05596...	0.029308...	0.124124...	0.0	0.0384615...	0.0340909...	16.0
17	0.21105...	0.029308...	0.045920...	0.0	0.0384615...	0.0227272...	16.0
18	0.21105...	0.029308...	0.030280...	0.0	0.0192307...	0.0113636...	16.0
19	0.39312...	0.022044...	0.030280...	0.0	0.0192307...	0.0056818...	16.0
20	0.52798...	0.006012...	0.124124...	0.0	0.0192307...	0.0227272...	16.0
21	0.02629...	0.029308...	0.061561...	0.0	0.0192307...	0.0340909...	17.0
22	0.04922...	0.006012...	0.014639...	0.0	0.0576923...	0.0568181...	17.0
23	0.10991...	0.006199...	0.061561...	0.0	0.0192307...	0.0170454...	18.0
24	0.53472...	0.014028...	0.007007...	0.03125	0.0192307...	0.0056818...	18.0

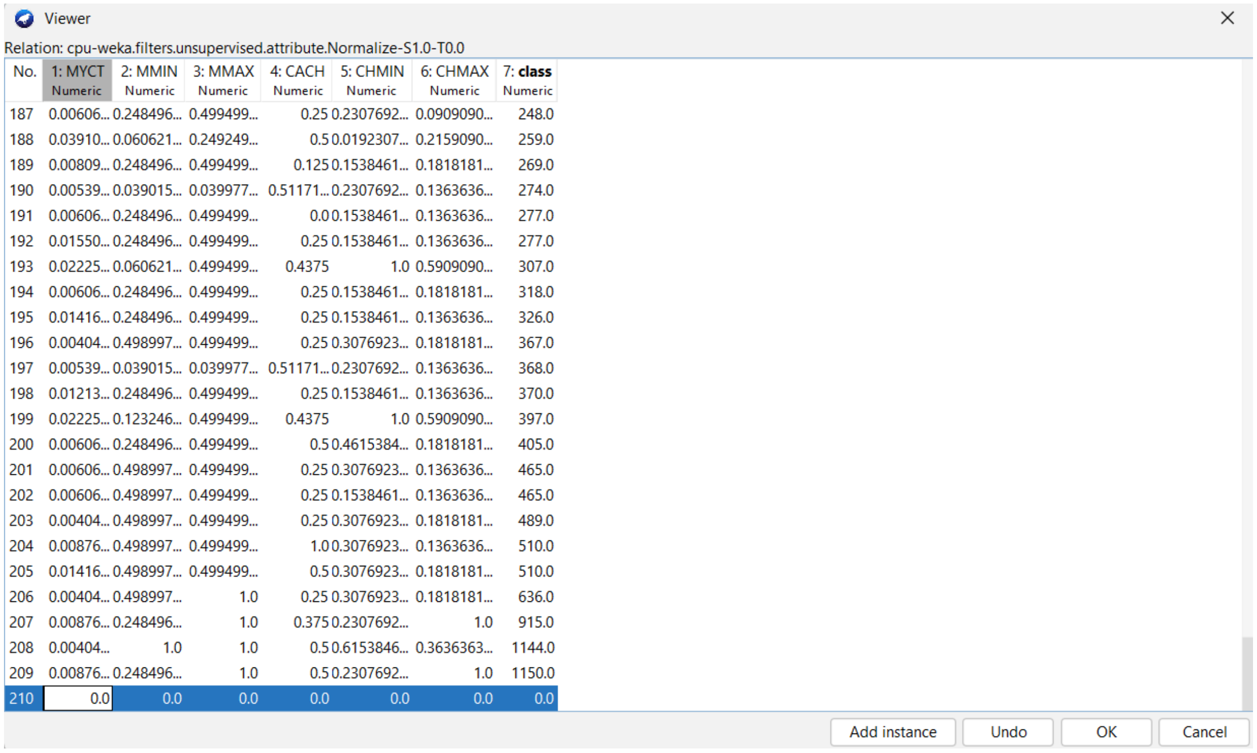
An instance can also be deleted by right clicking on the the value



Relation: cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

No.	1: MYCT Numeric	2: MMIN Numeric	3: MMAX Numeric	4: CACH Numeric	5: CHMIN Numeric	6: CHMAX Numeric	7: **class** Numeric
183	0.01550...	0.248496...	0.249249...	0.125	0.1538461...	0.0909090...	214.0
184	0.02090...	0.123246...	0.374374...	0.125	0.1538461...	0.136363	
185	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.181818	
186	0.02697...	0.123246...	0.374374...	0.25	0.2307692...	0.090909	
187	0.00606...	0.248496...	0.499499...	0.25	0.2307692...	0.090909	
188	0.03910...	0.060621...	0.249249...	0.5	0.0192307...	0.215909	
189	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.181818	
190	0.00539...	0.039015...	0.039977...	0.51171...	0.2307692...	0.136363	
191	0.00606...	0.248496...	0.499499...	0.0	0.1538461...	0.136363	
192	0.01550...	0.248496...	0.499499...	0.25	0.1538461...	0.136363	
193	0.02225...	0.060621...	0.499499...	0.4375	1.0	0.590909	
194	0.00606...	0.248496...	0.499499...	0.25	0.1538461...	0.181818	
195	0.01416...	0.248496...	0.499499...	0.25	0.1538461...	0.1363636...	326.0

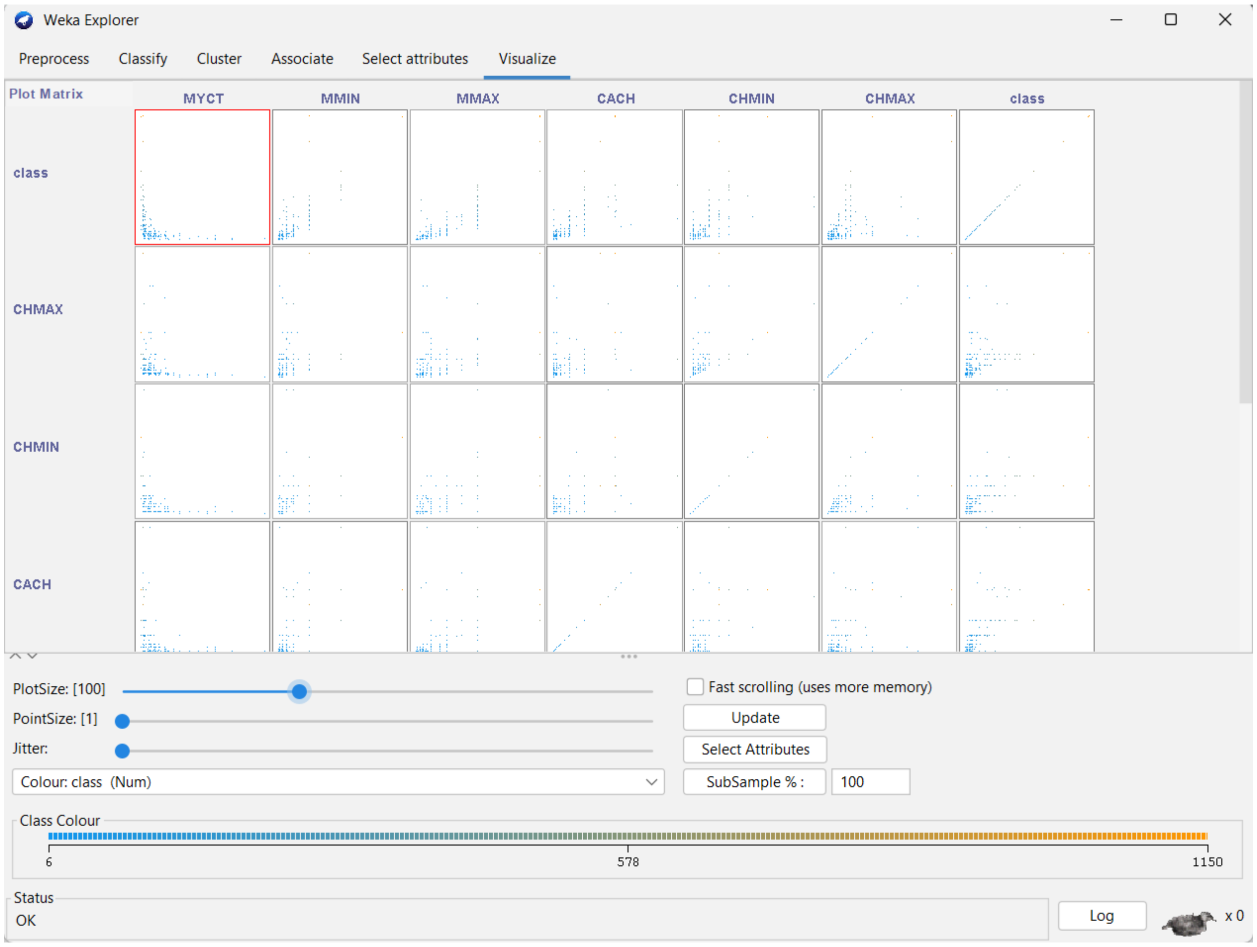
A new instance can also be added by clicking on the “Add instance” on the bottom.



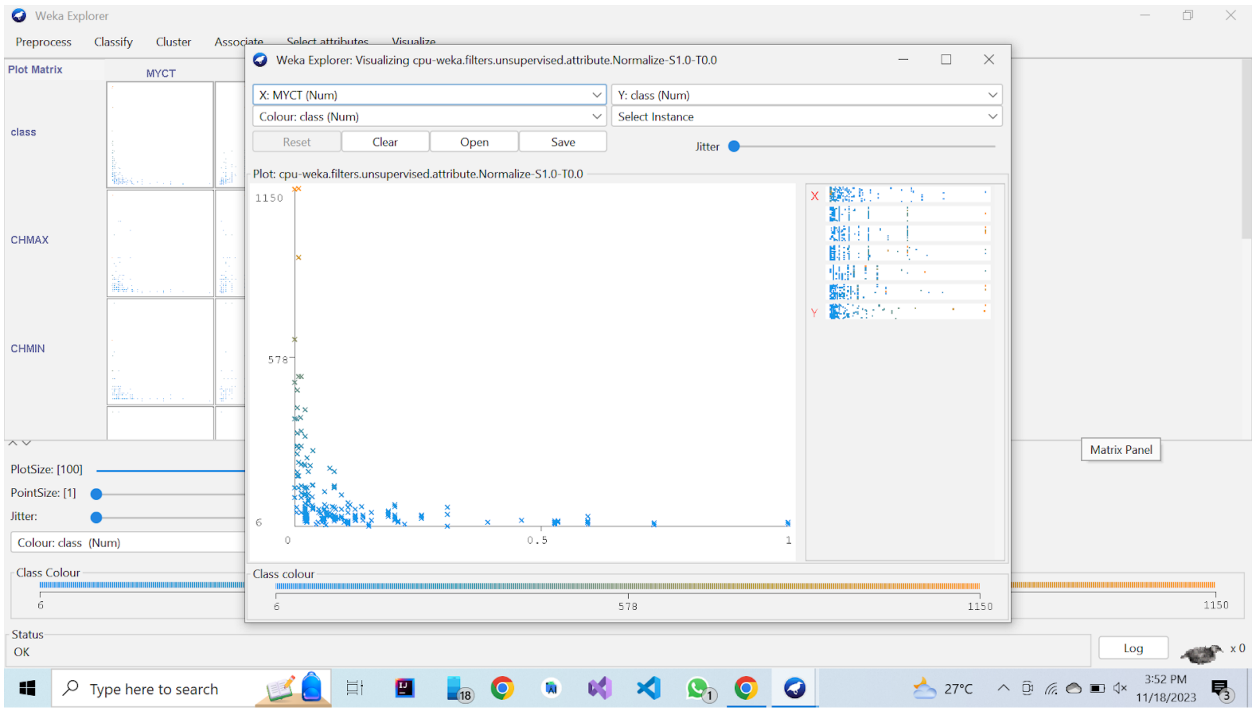
Relation: cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

No.	1: MYCT Numeric	2: MMIN Numeric	3: MMAX Numeric	4: CACH Numeric	5: CHMIN Numeric	6: CHMAX Numeric	7: **class** Numeric
187	0.00606...	0.248496...	0.499499...	0.25	0.2307692...	0.0909090...	248.0
188	0.03910...	0.060621...	0.249249...	0.5	0.0192307...	0.2159090...	259.0
189	0.00809...	0.248496...	0.499499...	0.125	0.1538461...	0.1818181...	269.0
190	0.00539...	0.039015...	0.039977...	0.51171...	0.2307692...	0.1363636...	274.0
191	0.00606...	0.248496...	0.499499...	0.0	0.1538461...	0.1363636...	277.0
192	0.01550...	0.248496...	0.499499...	0.25	0.1538461...	0.1363636...	277.0
193	0.02225...	0.060621...	0.499499...	0.4375	1.0	0.5909090...	307.0
194	0.00606...	0.248496...	0.499499...	0.25	0.1538461...	0.1818181...	318.0
195	0.01416...	0.248496...	0.499499...	0.25	0.1538461...	0.1363636...	326.0
196	0.00404...	0.498997...	0.499499...	0.25	0.3076923...	0.1818181...	367.0
197	0.00539...	0.039015...	0.039977...	0.51171...	0.2307692...	0.1363636...	368.0
198	0.01213...	0.248496...	0.499499...	0.25	0.1538461...	0.1363636...	370.0
199	0.02225...	0.123246...	0.499499...	0.4375	1.0	0.5909090...	397.0
200	0.00606...	0.248496...	0.499499...	0.5	0.4615384...	0.1818181...	405.0
201	0.00606...	0.498997...	0.499499...	0.25	0.3076923...	0.1363636...	465.0
202	0.00606...	0.498997...	0.499499...	0.25	0.1538461...	0.1363636...	465.0
203	0.00404...	0.498997...	0.499499...	0.25	0.3076923...	0.1818181...	489.0
204	0.00876...	0.498997...	0.499499...	1.0	0.3076923...	0.1363636...	510.0
205	0.01416...	0.498997...	0.499499...	0.5	0.3076923...	0.1818181...	510.0
206	0.00404...	0.498997...	1.0	0.25	0.3076923...	0.1818181...	636.0
207	0.00876...	0.248496...	1.0	0.375	0.2307692...	1.0	915.0
208	0.00404...	1.0	1.0	0.5	0.6153846...	0.3636363...	1144.0
209	0.00876...	0.248496...	1.0	0.5	0.2307692...	1.0	1150.0
210	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Step 6 :
The scatter plot can be visualized by going into the “Visualize” tab



PlotSize: [100]
PointSize: [1]
Jitter:
Colour: class (Num)
SubSample % : 100
Status
OK



Weka Explorer: Visualizing cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

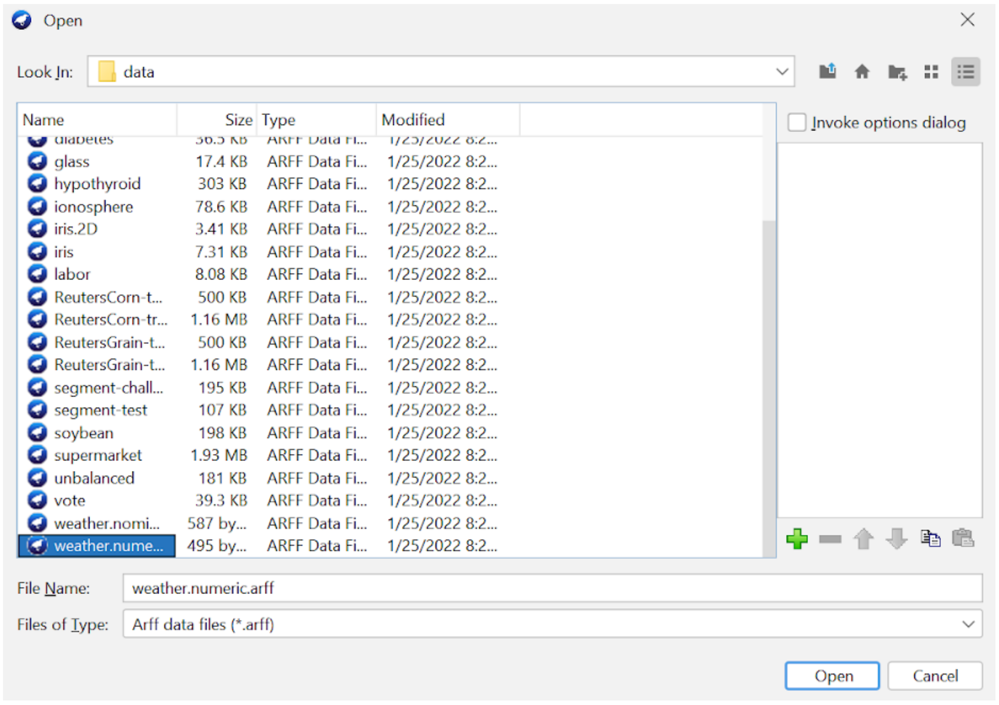
X: MYCT (Num)
Y: class (Num)
Colour: class (Num)
Select Instance

Plot: cpu-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0

PlotSize: [100]
PointSize: [1]
Jitter:
Colour: class (Num)
Status
OK

Practical 4
Implementation of any one classifier using JAVA and verify results with WEKA.

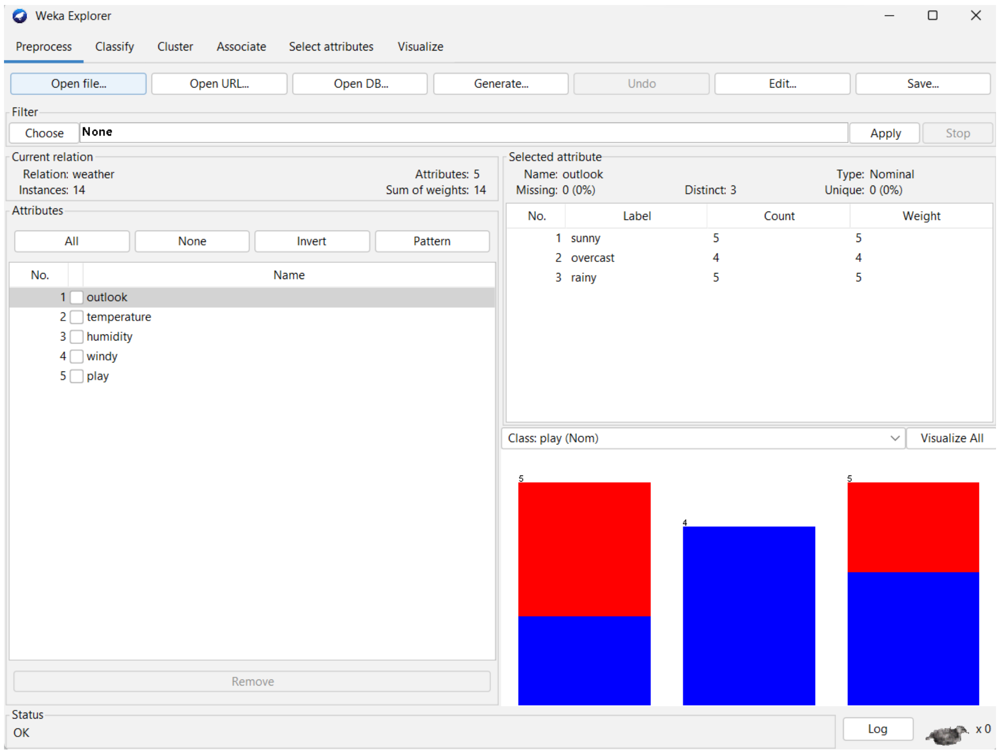
Step 1 :
Open the “Explorer” application, then “Open File” and choose the “weather.numeric” dataset.



Name	Size	Type	Modified
diabetes	36.5 KB	ARFF Data Fi...	1/25/2022 8:2...
glass	17.4 KB	ARFF Data Fi...	1/25/2022 8:2...
hypothyroid	303 KB	ARFF Data Fi...	1/25/2022 8:2...
ionosphere	78.6 KB	ARFF Data Fi...	1/25/2022 8:2...
iris.2D	3.41 KB	ARFF Data Fi...	1/25/2022 8:2...
iris	7.31 KB	ARFF Data Fi...	1/25/2022 8:2...
labor	8.08 KB	ARFF Data Fi...	1/25/2022 8:2...
ReutersCorn-t...	500 KB	ARFF Data Fi...	1/25/2022 8:2...
ReutersCorn-tr...	1.16 MB	ARFF Data Fi...	1/25/2022 8:2...
ReutersGrain-t...	500 KB	ARFF Data Fi...	1/25/2022 8:2...
ReutersGrain-t...	1.16 MB	ARFF Data Fi...	1/25/2022 8:2...
segment-chall...	195 KB	ARFF Data Fi...	1/25/2022 8:2...
segment-test	107 KB	ARFF Data Fi...	1/25/2022 8:2...
soybean	198 KB	ARFF Data Fi...	1/25/2022 8:2...
supermarket	1.93 MB	ARFF Data Fi...	1/25/2022 8:2...
unbalanced	181 KB	ARFF Data Fi...	1/25/2022 8:2...
vote	39.3 KB	ARFF Data Fi...	1/25/2022 8:2...
weather.nomi...	587 by...	ARFF Data Fi...	1/25/2022 8:2...
weather.nume...	495 by...	ARFF Data Fi...	1/25/2022 8:2...

File Name: weather.numeric.arff

Files of Type: Arff data files (*.arff)



Filter: None

Current relation
Relation: weather   Attributes: 5
Instances: 14   Sum of weights: 14

Attributes

No.	Name
1	outlook
2	temperature
3	humidity
4	windy
5	play

Selected attribute
Name: outlook   Type: Nominal
Missing: 0 (0%)   Distinct: 3   Unique: 0 (0%)

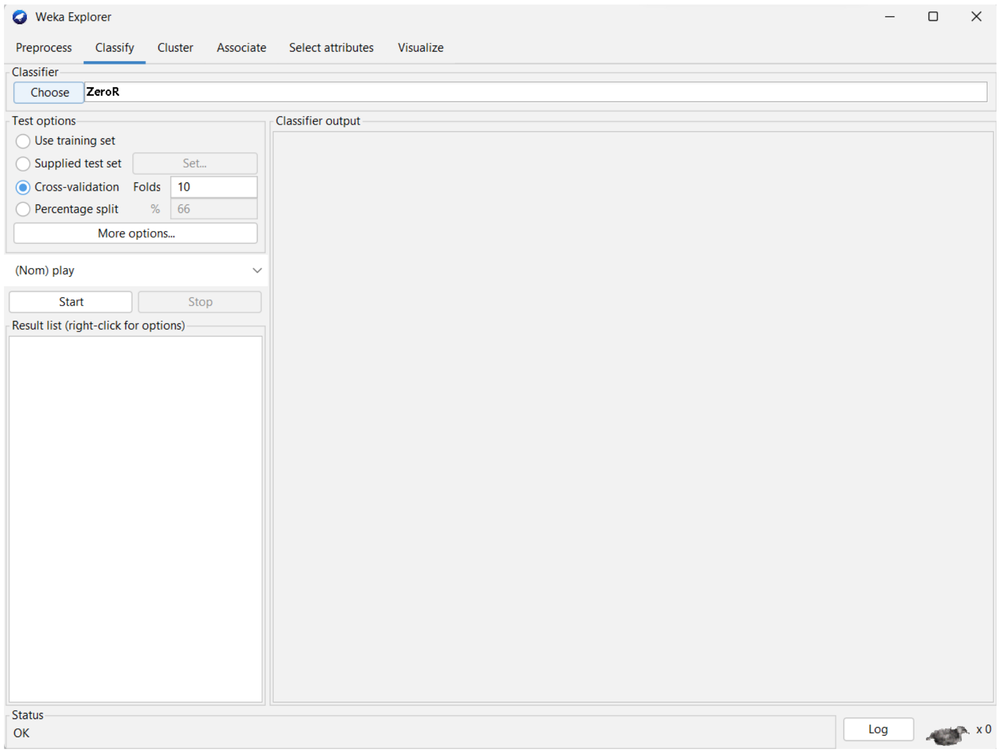
No.	Label	Count	Weight
1	sunny	5	5
2	overcast	4	4
3	rainy	5	5

Class: play (Nom)

Status
OK

Step 2 :

Go to the “Classify” tab



Classifier: ZeroR

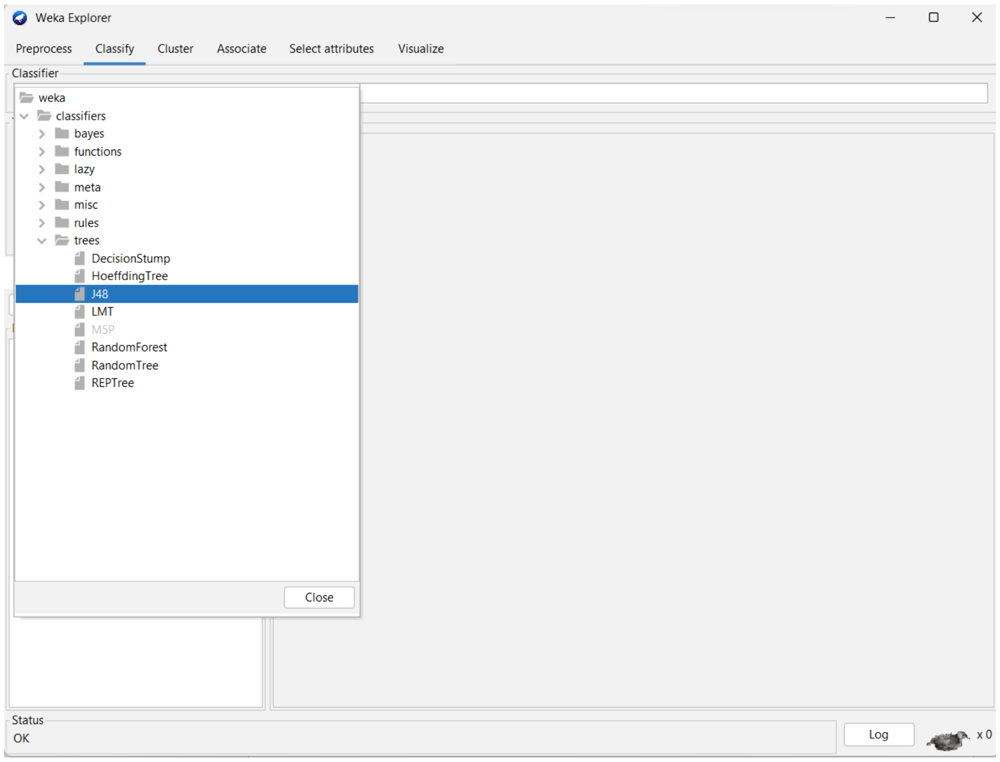
Cross-validation Folds 10

Percentage split % 66

(Nom) play

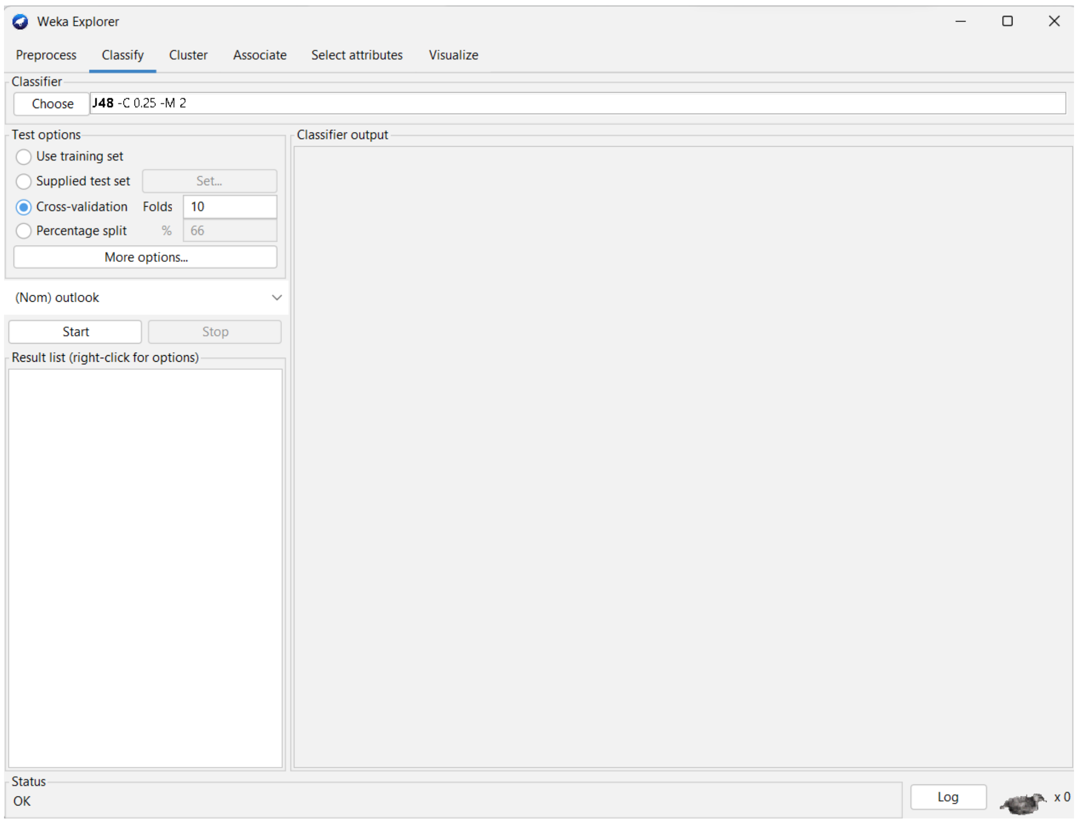
Status
OK

Click on “Choose” then select “J48” from “trees”.



Status
OK

Select the column required, then click on “Start”. Here. the column chosen is “outlook”



Classifier
J48 -C 0.25 -M 2

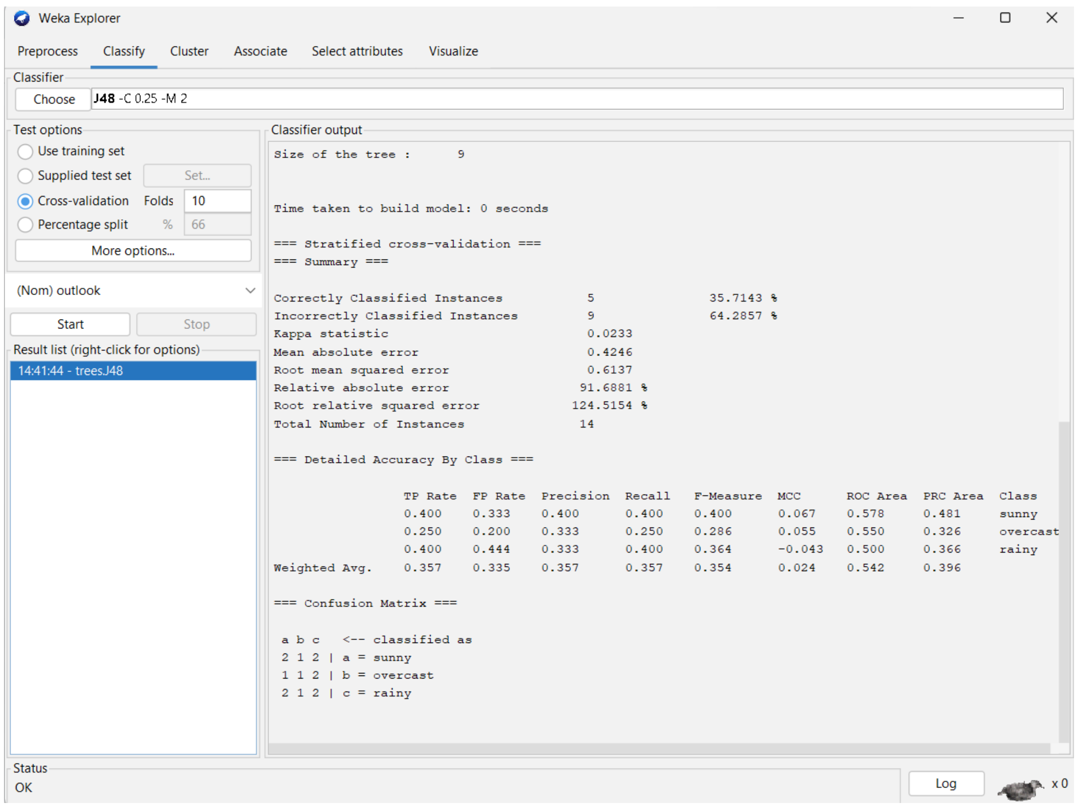
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66

(Nom) outlook

Result list (right-click for options)

Classifier output

Status
OK



Classifier
J48 -C 0.25 -M 2

Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66

(Nom) outlook

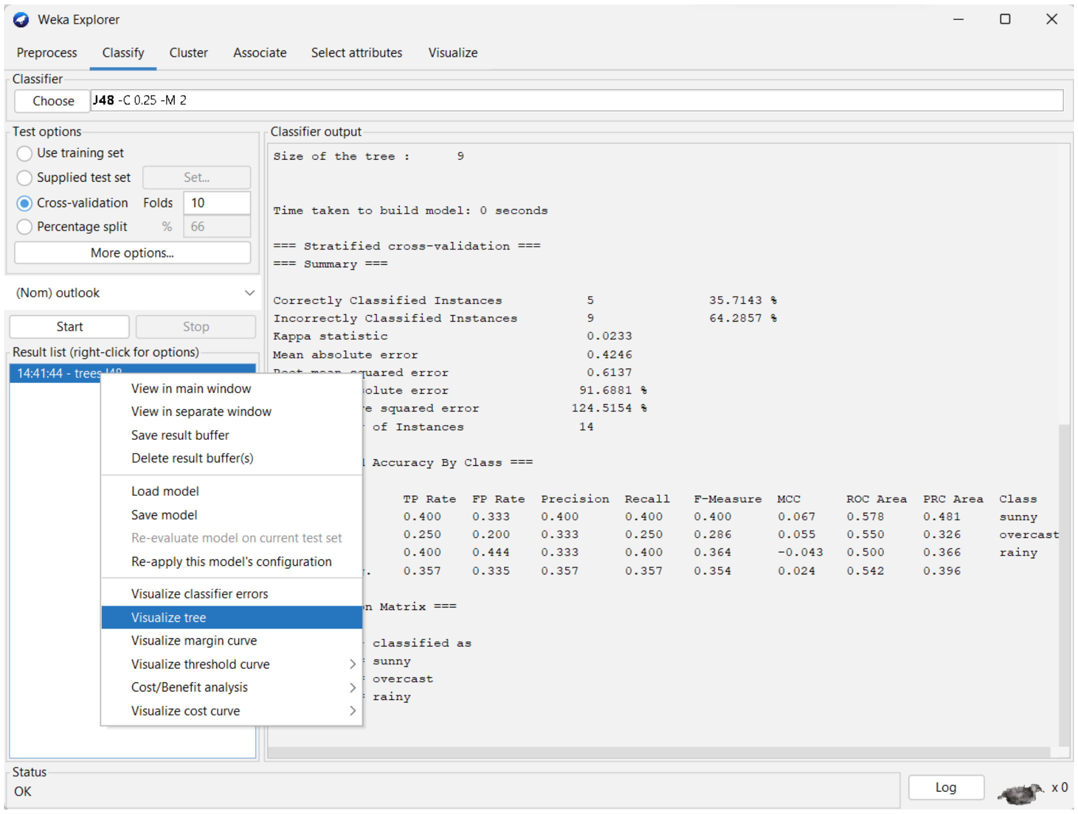
Result list (right-click for options)
14:41:44 - trees.J48

Classifier output

```
Size of the tree :      9


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           5               35.7143 %
Incorrectly Classified Instances         9               64.2857 %
Kappa statistic                          0.0233
Mean absolute error                      0.4246
Root mean squared error                  0.6137
Relative absolute error                 91.6881 %
Root relative squared error            124.5154 %
Total Number of Instances               14

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.400    0.333    0.400      0.400    0.400      0.067    0.578     0.481     sunny
                 0.250    0.200    0.333      0.250    0.286      0.055    0.550     0.326     overcast
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.500     0.366     rainy
Weighted Avg.    0.357    0.335    0.357      0.357    0.354      0.024    0.542     0.396

=== Confusion Matrix ===

 a b c   <-- classified as
 2 1 2 | a = sunny
 1 1 2 | b = overcast
 2 1 2 | c = rainy
```

Status
OK

Right click on the J48 tab and select “Visualize tree”



Weka Explorer

Preprocess | Classify | Cluster | Associate | Select attributes | Visualize

Classifier: Choose **J48** -C 0.25 -M 2

Test options:
- Use training set
- Supplied test set
- [✓ Cross-validation] Folds 10
- Percentage split % 66

(Nom) outlook

Result list (right-click for options): 14:41:44 - trees.J48

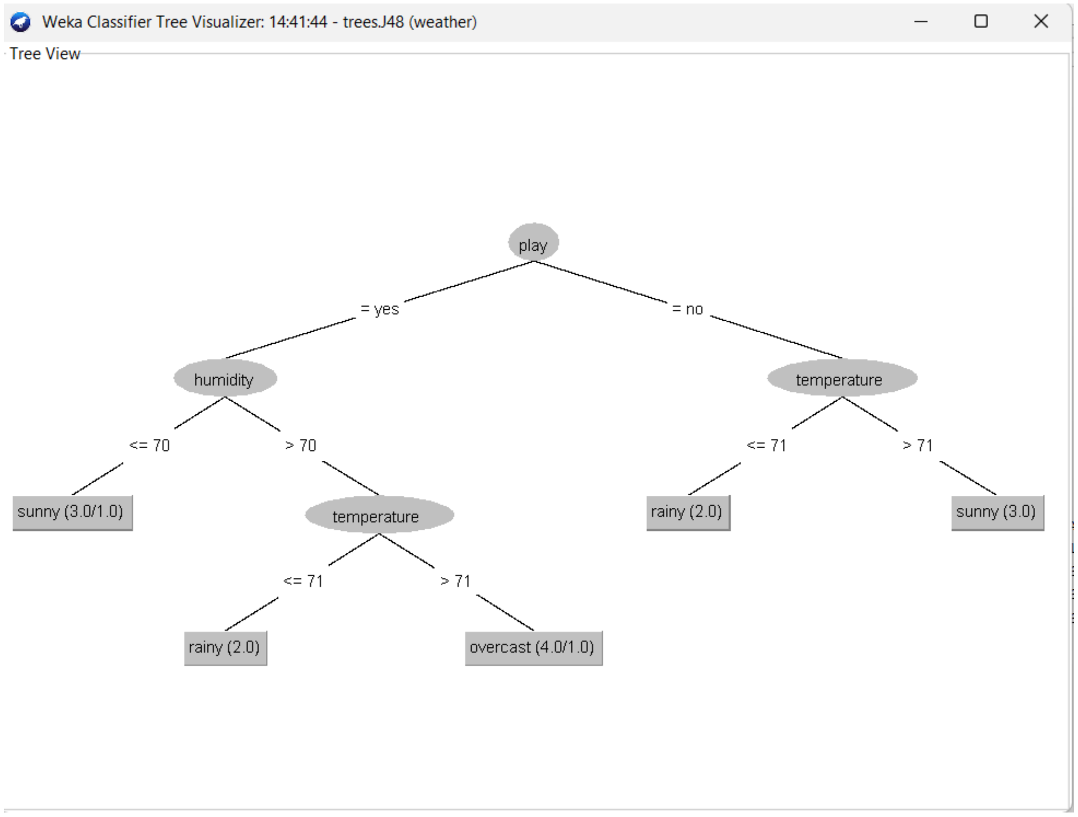
Classifier output

```
Size of the tree :      9


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances           5               35.7143 %
Incorrectly Classified Instances         9               64.2857 %
Kappa statistic                          0.0233
Mean absolute error                      0.4246
Root mean squared error                  0.6137
            olute error                 91.6881 %
            e squared error            124.5154 %
            of Instances                14

            Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.400    0.333    0.400      0.400    0.400      0.067    0.578     0.481     sunny
                 0.250    0.200    0.333      0.250    0.286      0.055    0.550     0.326     overcast
                 0.400    0.444    0.333      0.400    0.364      -0.043   0.500     0.366     rainy
            .    0.357    0.335    0.357      0.357    0.354      0.024    0.542     0.396

            n Matrix ===

            classified as
            sunny
            overcast
            rainy
```

Status: OK

Weka Classifier Tree Visualizer: 14:41:44 - trees.J48 (weather)

Tree View

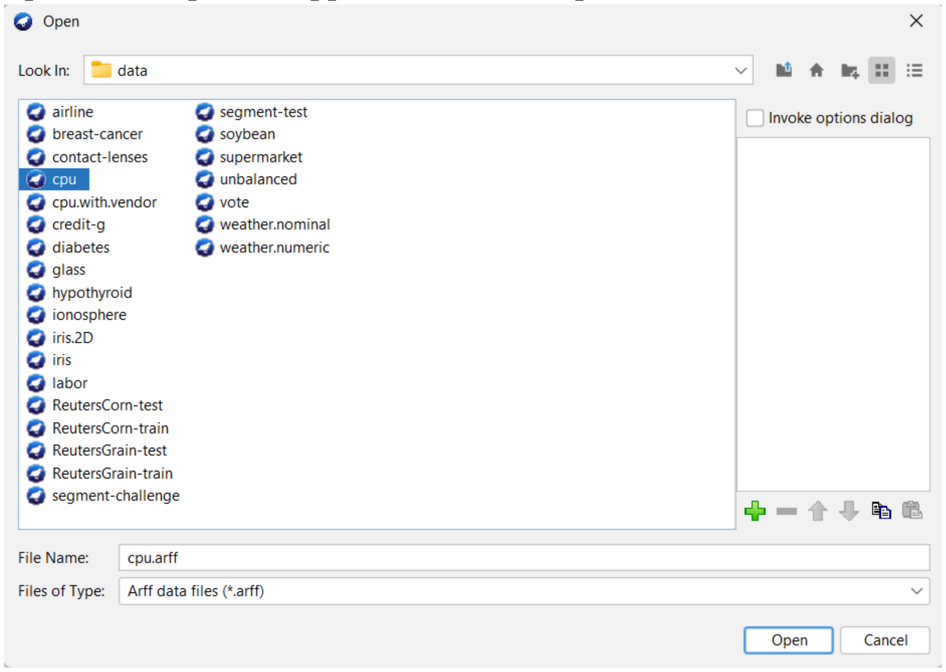


Practical 5

Implementation of any one clustering algorithm using JAVA and verify results with WEKA.

Step 1 :

Open the “Explorer” application, then “Open File” and choose the “cpu” dataset.

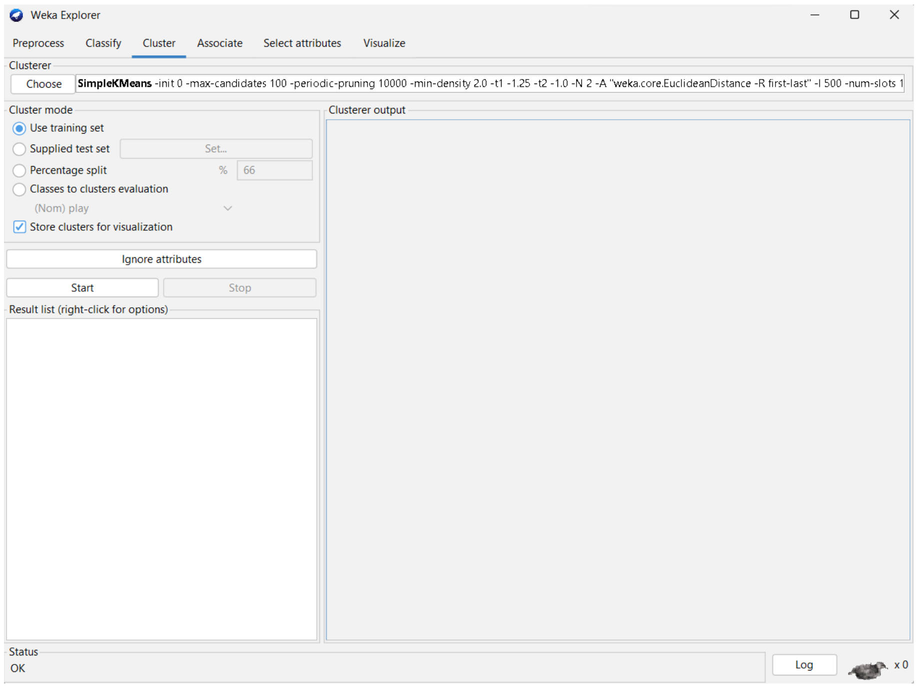


File Name: cpu.arff

Files of Type: Arff data files (*.arff)

Step 2 :

Go to the “Cluster” tab

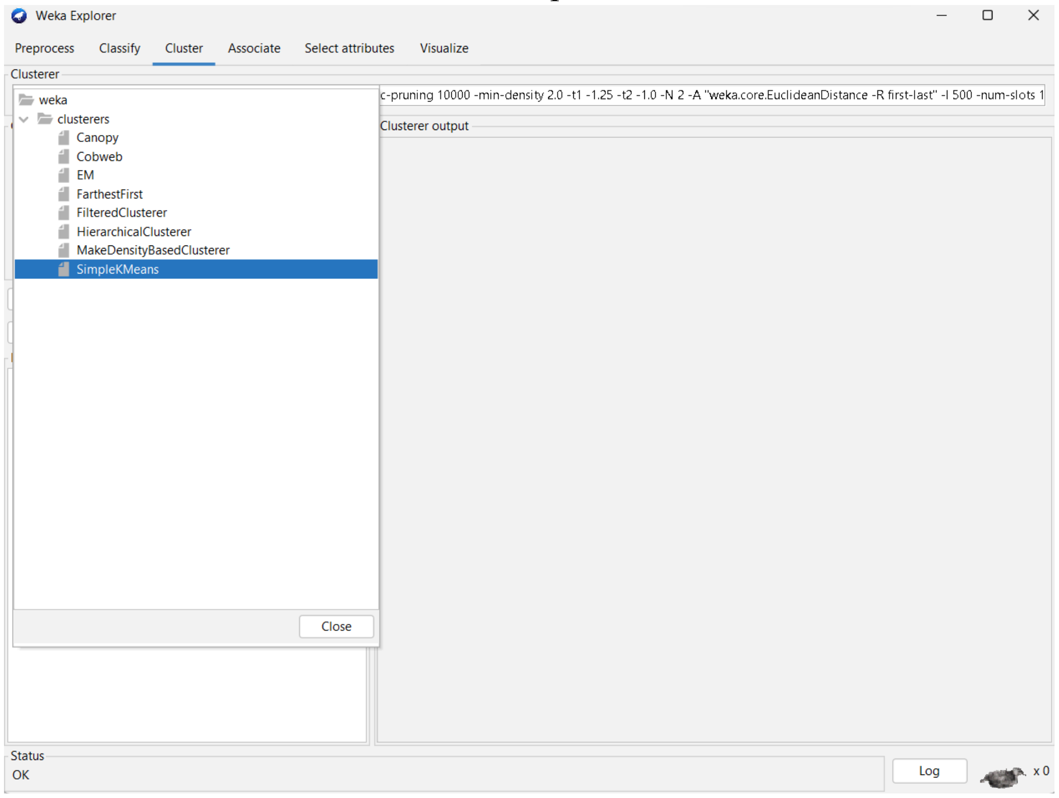


Clusterer: SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1

Cluster mode:
- [✓ Use training set]
- Supplied test set
- Percentage split % 66
- Classes to clusters evaluation
- (Nom) play
- [✓] Store clusters for visualization

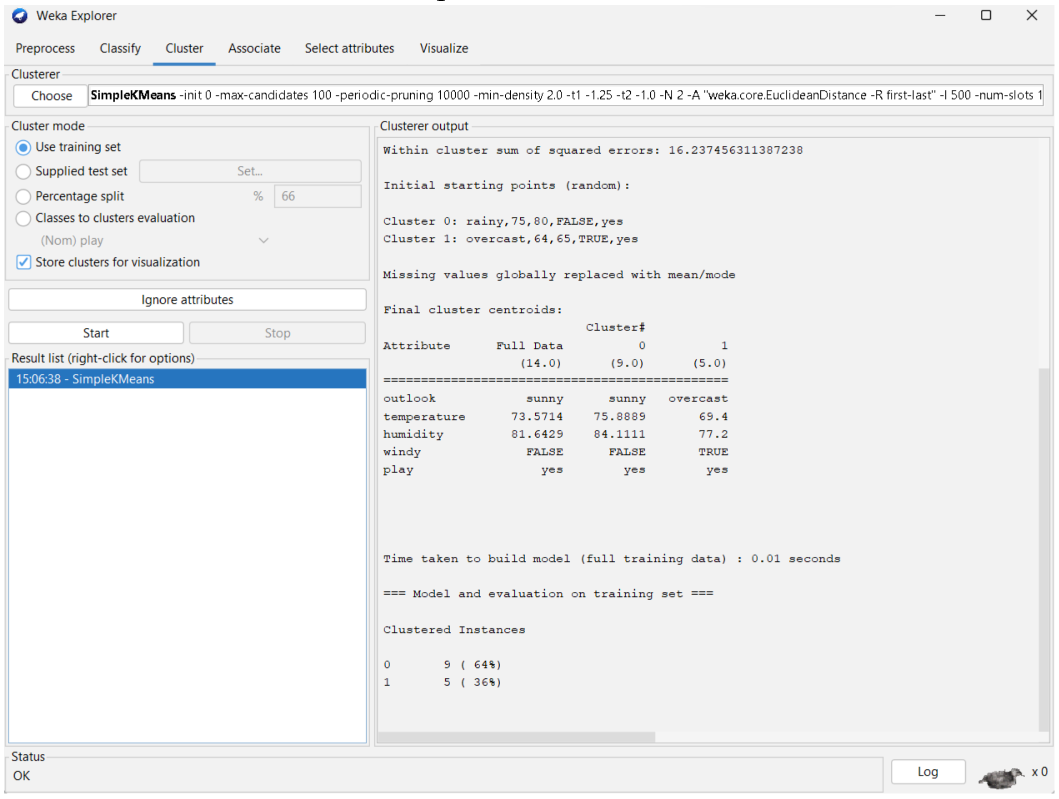
Status: OK

Click on “Choose” and then select “simpleKMeans”



c-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

Then, click on “ Start”. The output is shown.

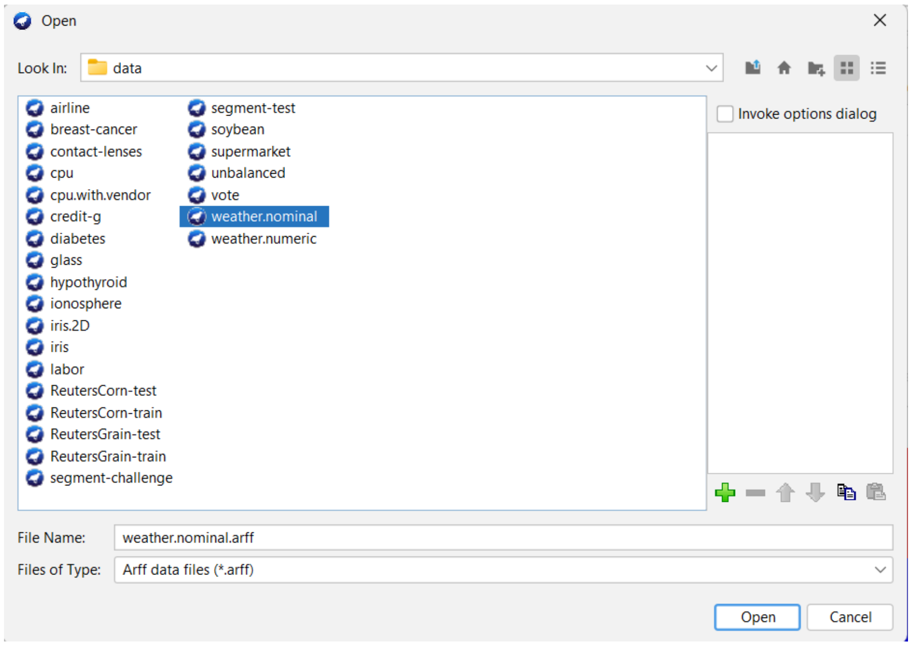


SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1

```
Within cluster sum of squared errors: 16.237456311387238

Initial starting points (random):

Cluster 0: rainy,75,80,FALSE,yes
Cluster 1: overcast,64,65,TRUE,yes

Missing values globally replaced with mean/mode

Final cluster centroids:
                           Cluster#
Attribute      Full Data          0          1
                  (14.0)      (9.0)      (5.0)
==============================================
outlook            sunny      sunny   overcast
temperature      73.5714    75.8889       69.4
humidity         81.6429    84.1111       77.2
windy              FALSE      FALSE       TRUE
play                 yes        yes        yes




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

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

Clustered Instances

0       9 ( 64%)
1       5 ( 36%)
```

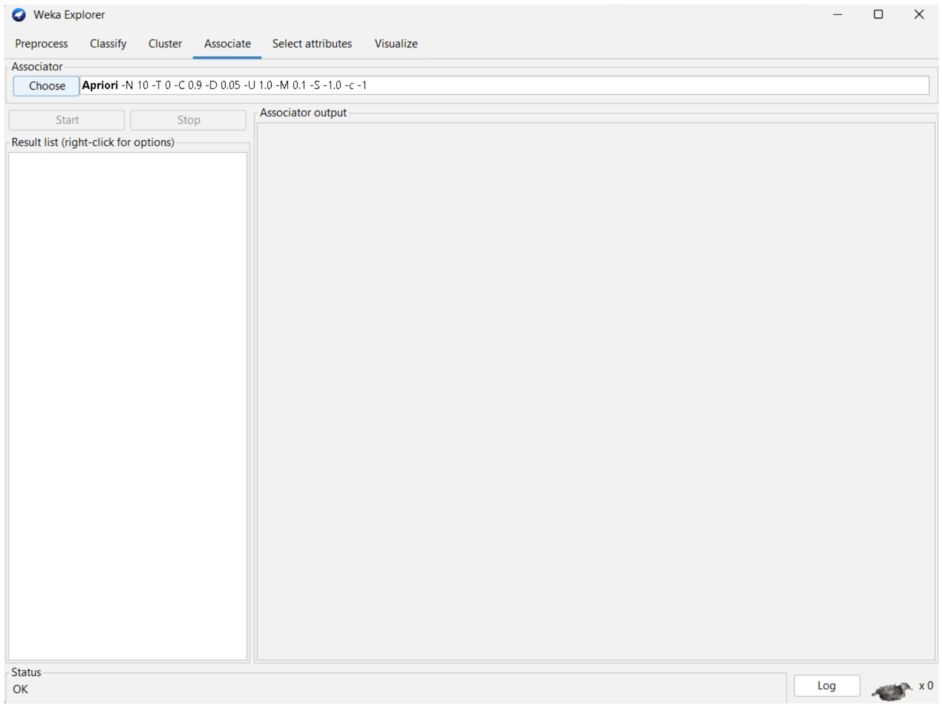
Practical 6
Implementation of association mining rule –Apriori algorithm using JAVA and verify the result with WEKA.

Step 1 :
Open the “Explorer” application, then “Open File” and choose the “weather.nominal” dataset.



File Name: weather.nominal.arff
Files of Type: Arff data files (*.arff)

Step 2 :
Go to the “Associate” tab

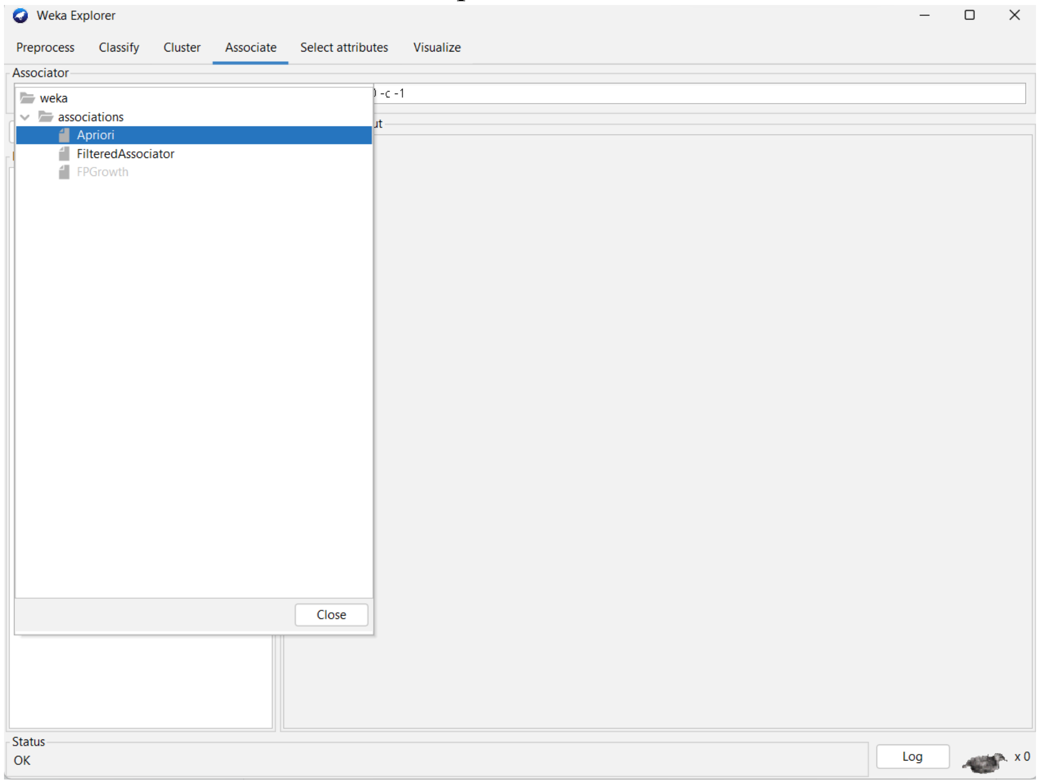


Associator: Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

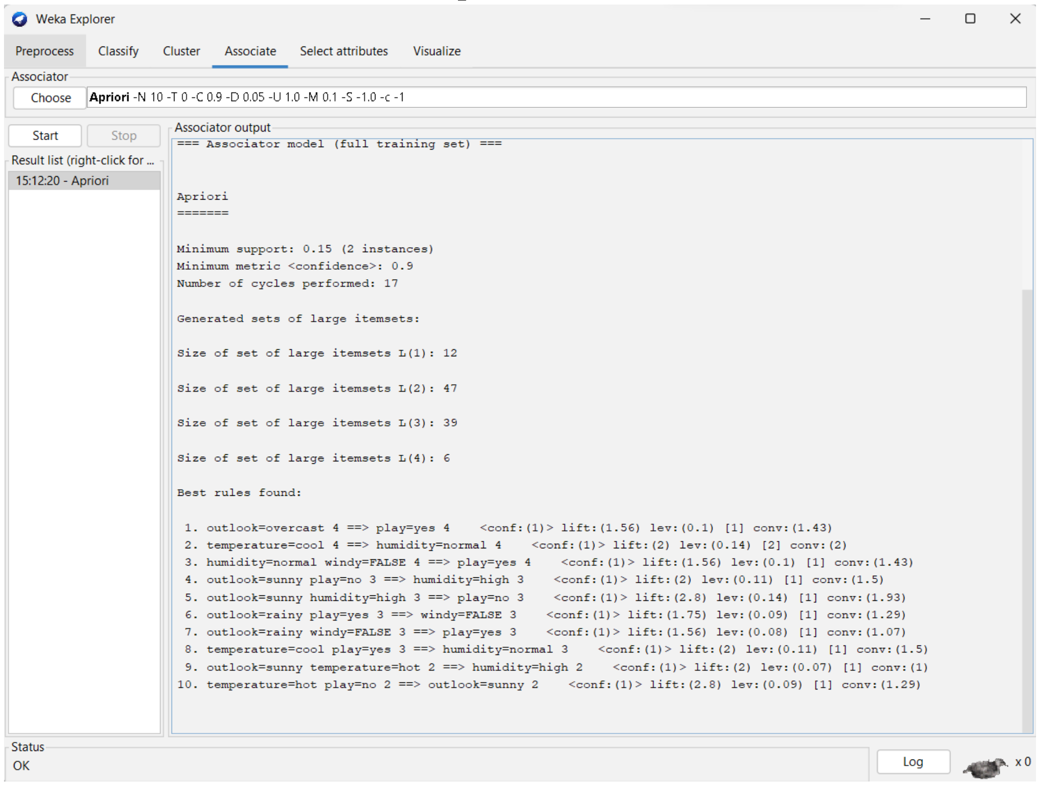
Status: OK

Step 3 :

Click on “Choose” and select “Apriori”



Then click on “Start” and the output is shown.

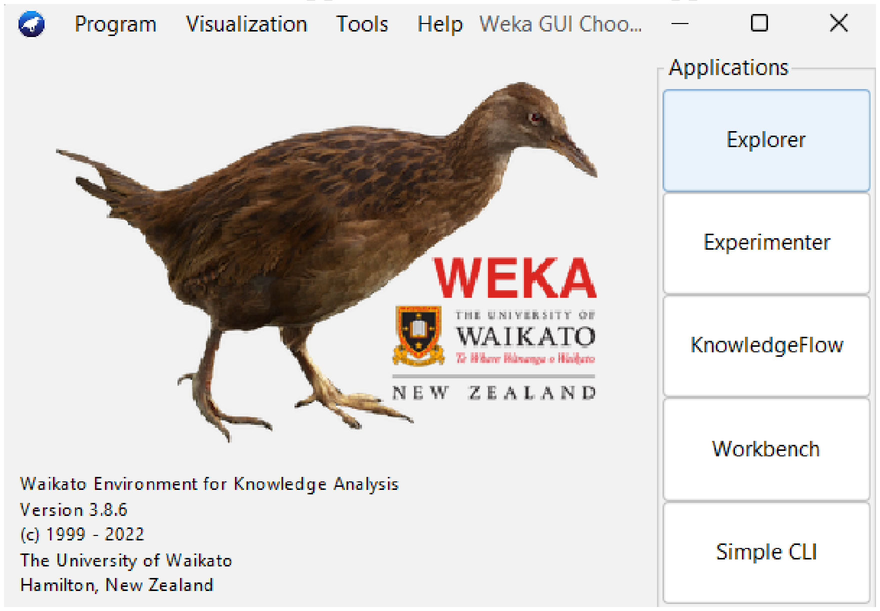


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

```
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.15 (2 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 12

Size of set of large itemsets L(2): 47

Size of set of large itemsets L(3): 39

Size of set of large itemsets L(4): 6

Best rules found:

 1. outlook=overcast 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 2. temperature=cool 4 ==> humidity=normal 4    <conf:(1)> lift:(2) lev:(0.14) [2] conv:(2)
 3. humidity=normal windy=FALSE 4 ==> play=yes 4    <conf:(1)> lift:(1.56) lev:(0.1) [1] conv:(1.43)
 4. outlook=sunny play=no 3 ==> humidity=high 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 5. outlook=sunny humidity=high 3 ==> play=no 3    <conf:(1)> lift:(2.8) lev:(0.14) [1] conv:(1.93)
 6. outlook=rainy play=yes 3 ==> windy=FALSE 3    <conf:(1)> lift:(1.75) lev:(0.09) [1] conv:(1.29)
 7. outlook=rainy windy=FALSE 3 ==> play=yes 3    <conf:(1)> lift:(1.56) lev:(0.08) [1] conv:(1.07)
 8. temperature=cool play=yes 3 ==> humidity=normal 3    <conf:(1)> lift:(2) lev:(0.11) [1] conv:(1.5)
 9. outlook=sunny temperature=hot 2 ==> humidity=high 2    <conf:(1)> lift:(2) lev:(0.07) [1] conv:(1)
10. temperature=hot play=no 2 ==> outlook=sunny 2    <conf:(1)> lift:(2.8) lev:(0.09) [1] conv:(1.29)
```

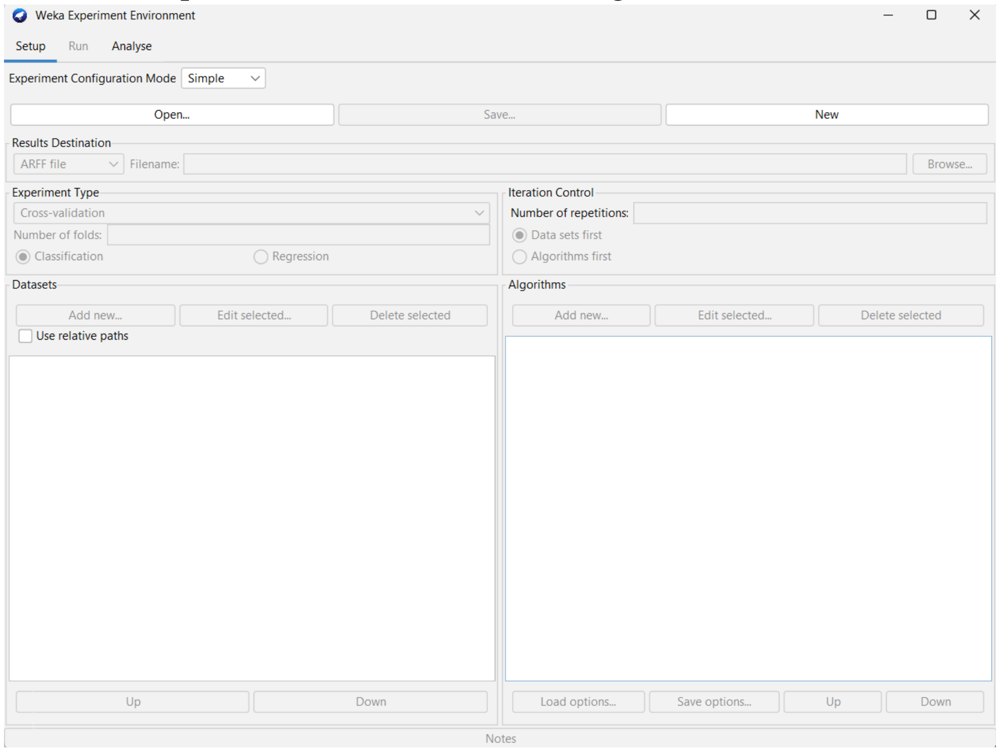
Practical 7
Using WEKA to compare different classifiers using Experimenter.

Step 1 :
Open the WEKA application and the first page is displayed where you have the option to choose from various applications that WEKA supports.



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Step 2 :
Click on "Experimenter" and the below image is shown



Experiment Configuration Mode: Simple

Results Destination: ARFF file, Filename:

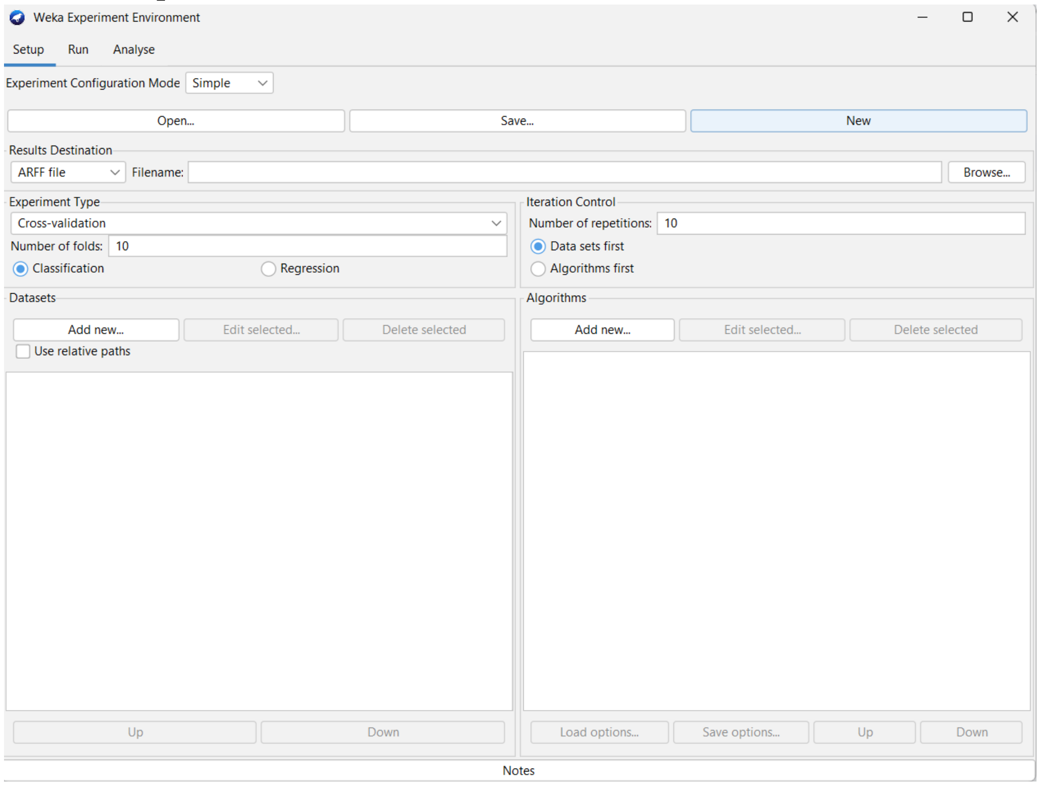
Experiment Type: Cross-validation
Number of folds:
[✓ Classification] Regression

Iteration Control
Number of repetitions:
[✓ Data sets first] Algorithms first

Datasets
Use relative paths

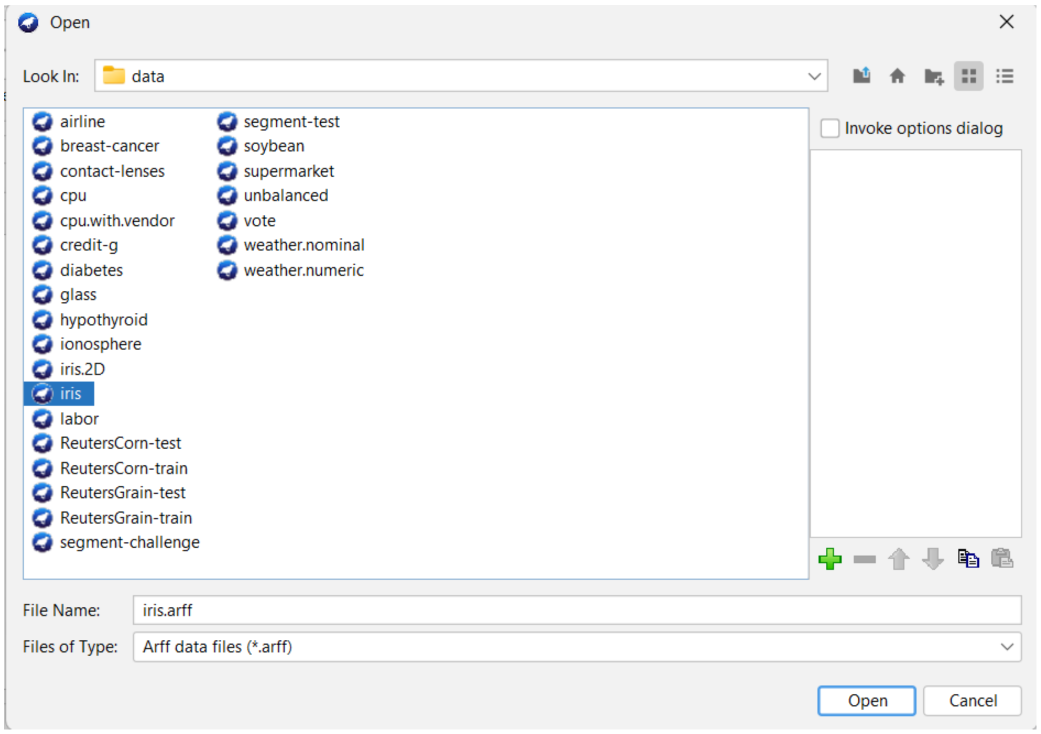
Algorithms

Step 3 :
Under Setup, click on "New"



Experiment Configuration Mode: Simple
Results Destination: ARFF file
Filename:
Experiment Type: Cross-validation
Number of folds: 10
[✓ Classification] Regression
Iteration Control
Number of repetitions: 10
[✓ Data sets first] Algorithms first
Datasets
Use relative paths
Algorithms

Step 4 :
Under "Datasets", select "Add new…". Then select the "iris" dataset.

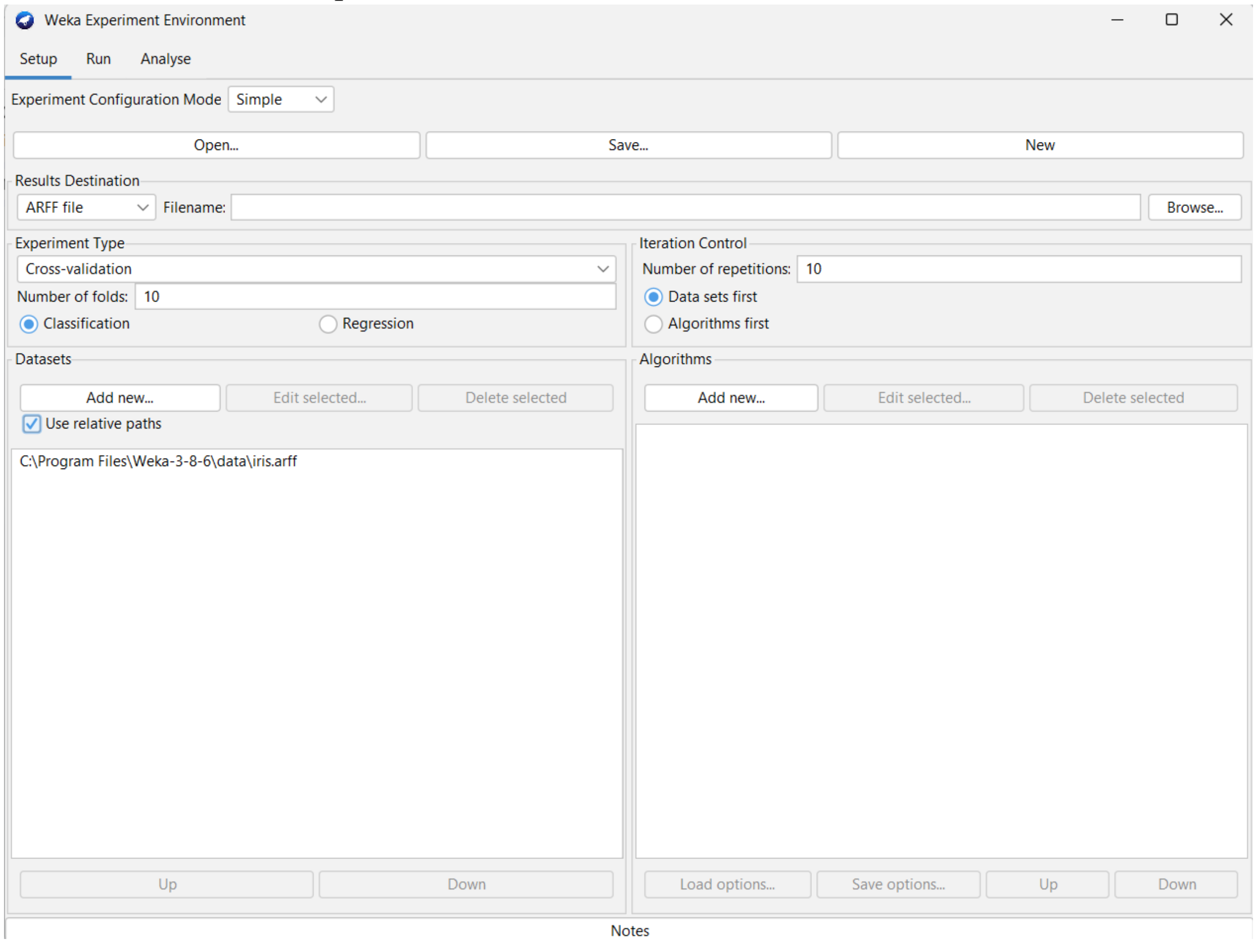


Look In: data

airline, breast-cancer, contact-lenses, cpu, cpu.with.vendor, credit-g, diabetes, glass, hypothyroid, ionosphere, iris.2D, iris, labor, ReutersCorn-test, ReutersCorn-train, ReutersGrain-test, ReutersGrain-train, segment-challenge, segment-test, soybean, supermarket, unbalanced, vote, weather.nominal, weather.numeric

File Name: iris.arff
Files of Type: Arff data files (*.arff)

Select the “Use relative paths” checkbox



Experiment Configuration Mode: Simple

Results Destination: ARFF file, Filename:

Experiment Type: Cross-validation

Number of folds: 10

Classification [✓] Regression

Iteration Control: Number of repetitions: 10

Data sets first [✓] Algorithms first

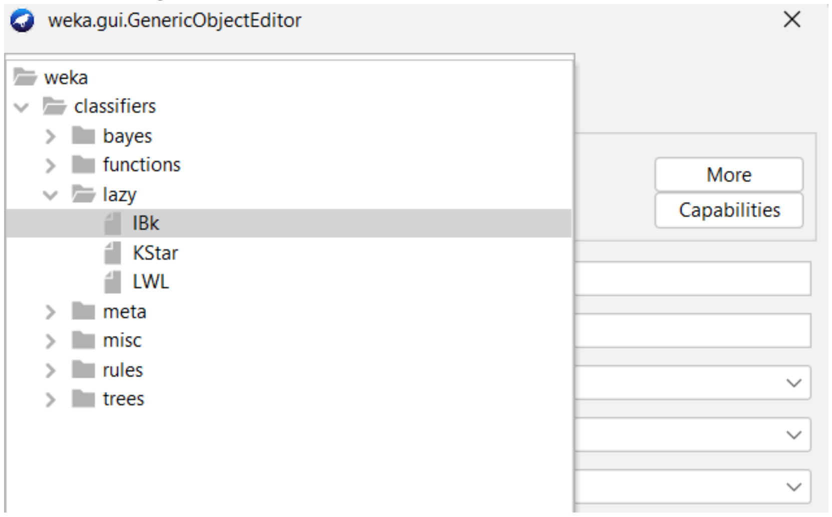
Datasets: Use relative paths [✓]

C:\Program Files\Weka-3-8-6\data\iris.arff

Algorithms

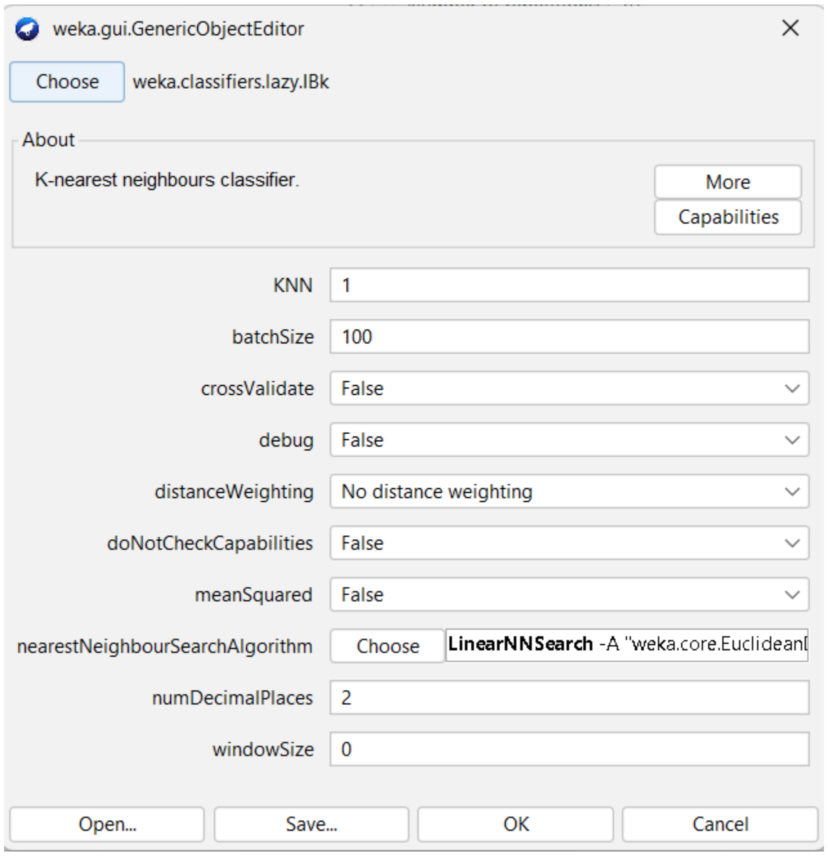
Step 5 :

Under the Algorithms, click on “Add new…”. Then click on Choose to select the different algorithms.



Under the "lazy" classifier, Select

i. IBk


weka.gui.GenericObjectEditor

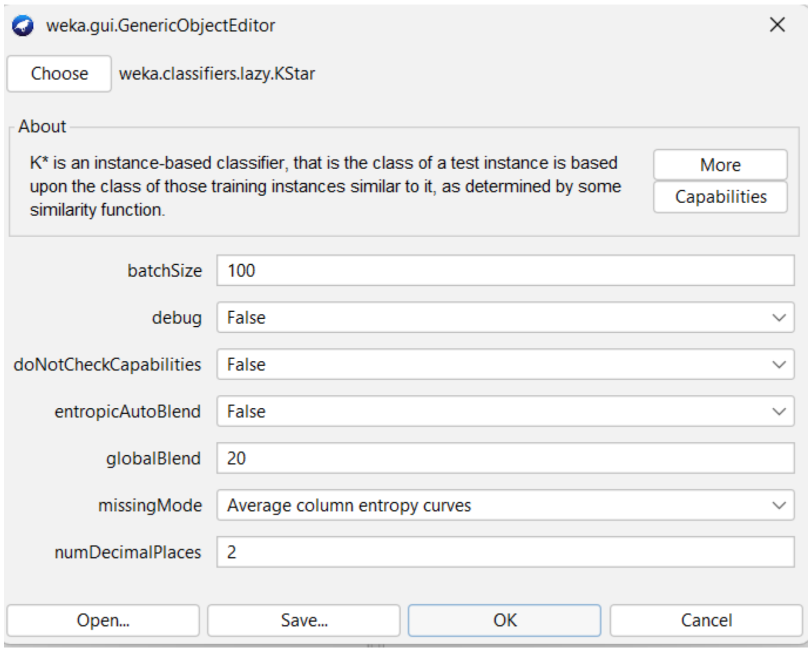
Choose weka.classifiers.lazy.IBk

About

K-nearest neighbours classifier.

KNN	1
batchSize	100
crossValidate	False
debug	False
distanceWeighting	No distance weighting
doNotCheckCapabilities	False
meanSquared	False
nearestNeighbourSearchAlgorithm	Choose **LinearNNSearch** -A "weka.core.EuclideanI
numDecimalPlaces	2
windowSize	0

ii. KStar


weka.gui.GenericObjectEditor

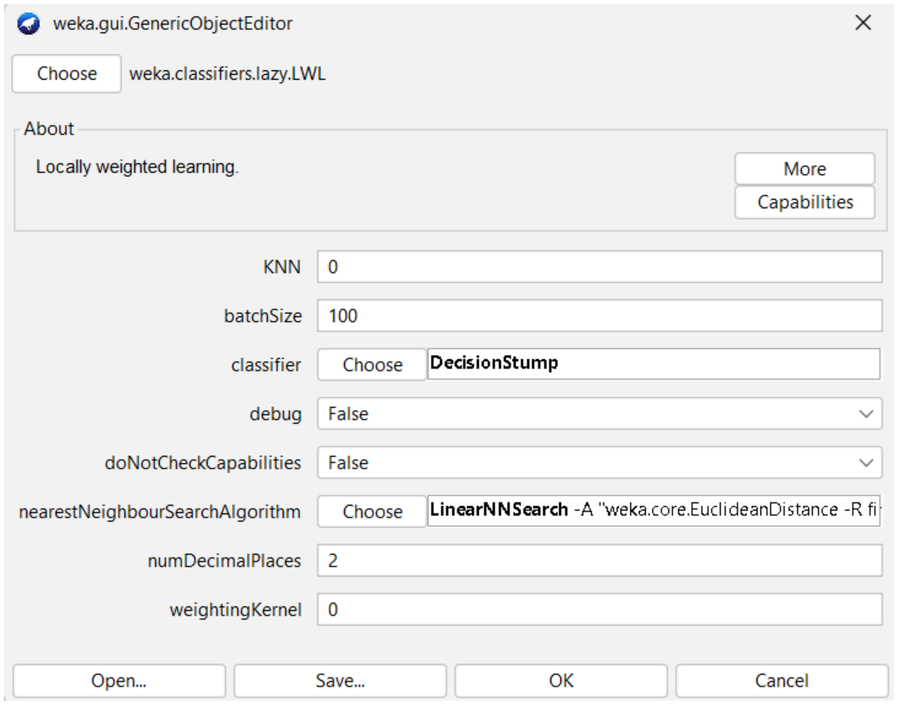
Choose weka.classifiers.lazy.KStar

About

K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function.

batchSize	100
debug	False
doNotCheckCapabilities	False
entropicAutoBlend	False
globalBlend	20
missingMode	Average column entropy curves
numDecimalPlaces	2

iii. LWL


weka.gui.GenericObjectEditor

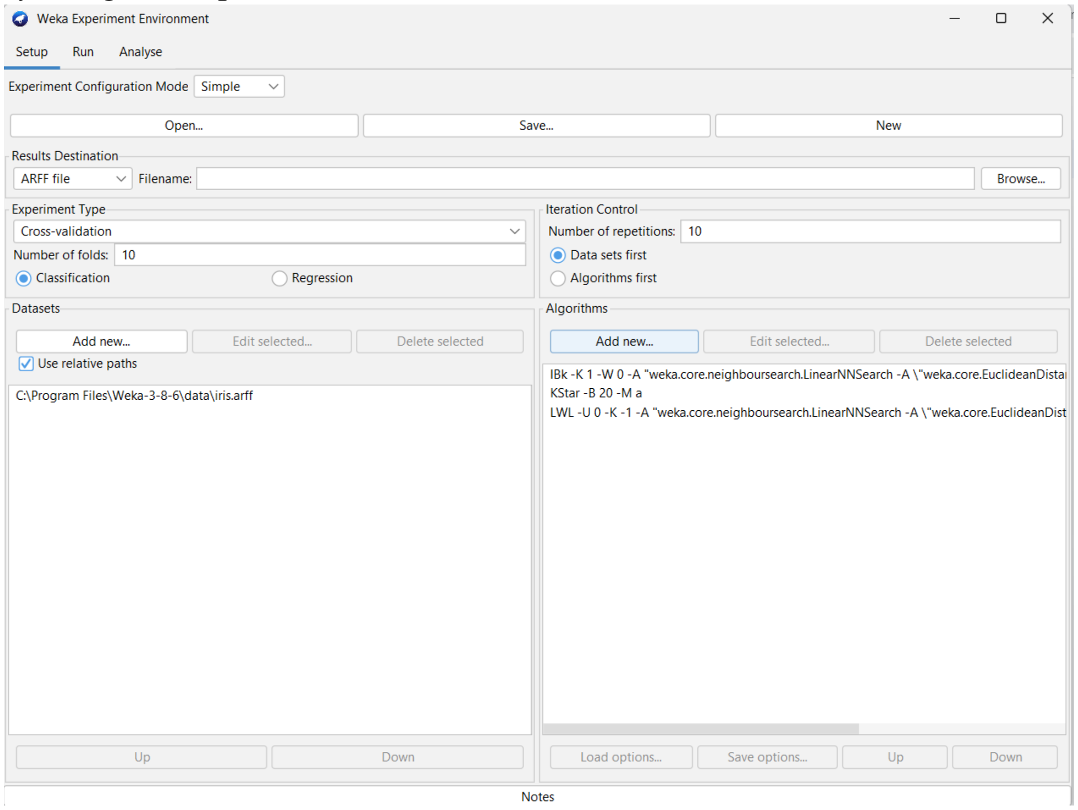
Choose weka.classifiers.lazy.LWL

About

Locally weighted learning.

KNN: 0
batchSize: 100
classifier: DecisionStump
debug: False
doNotCheckCapabilities: False
nearestNeighbourSearchAlgorithm: LinearNNSearch -A "weka.core.EuclideanDistance -R fi
numDecimalPlaces: 2
weightingKernel: 0

The screen is displayed below. We can also select each Algorithm and change the order by using the "Up" and "Down" function


Weka Experiment Environment

Experiment Configuration Mode: Simple

Results Destination: ARFF file Filename:

Experiment Type: Cross-validation
Number of folds: 10
[✓ Classification] Regression

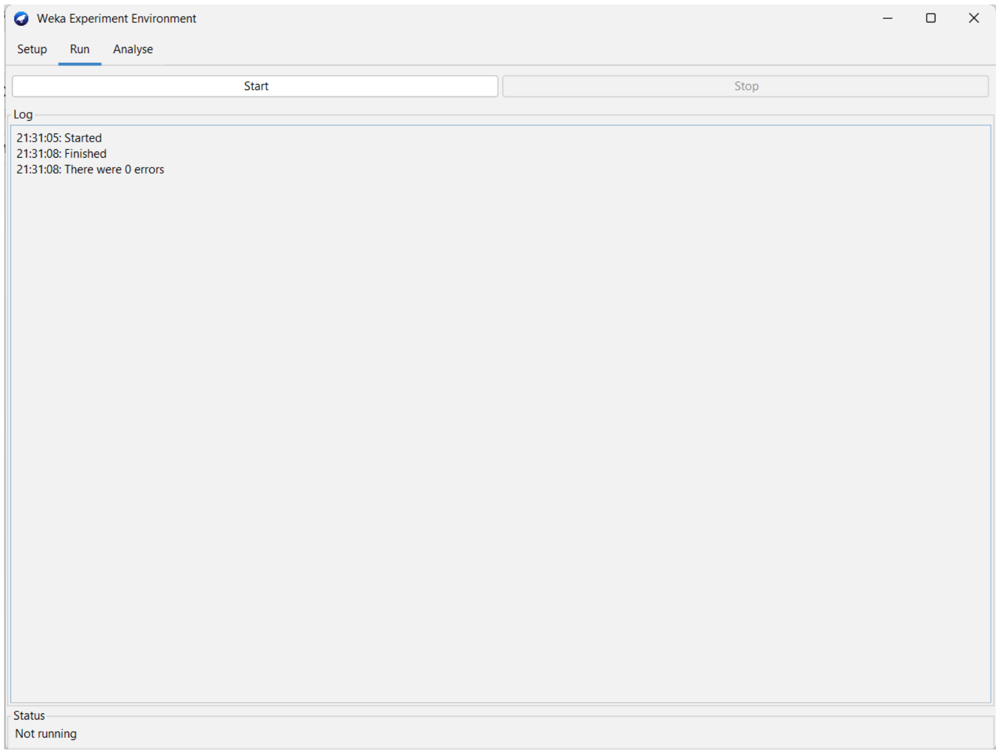
Iteration Control
Number of repetitions: 10
[✓ Data sets first] Algorithms first

Datasets
[✓ Use relative paths]
C:\Program Files\Weka-3-8-6\data\iris.arff

Algorithms
IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDista
KStar -B 20 -M a
LWL -U 0 -K -1 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDist

Step 6 :

Go to the “Run” tab and click on “Start”

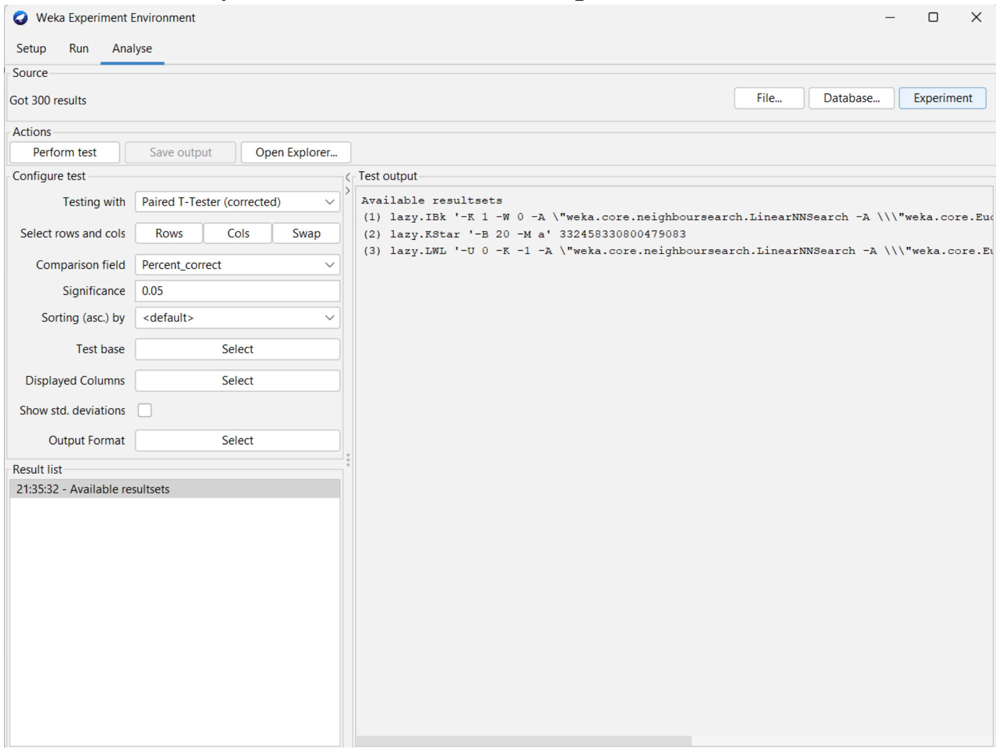


```
21:31:05: Started
21:31:08: Finished
21:31:08: There were 0 errors
```

Not running

Step 7 :

Go to the “Analyze” tab and click on “Experiment”



Got 300 results

Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

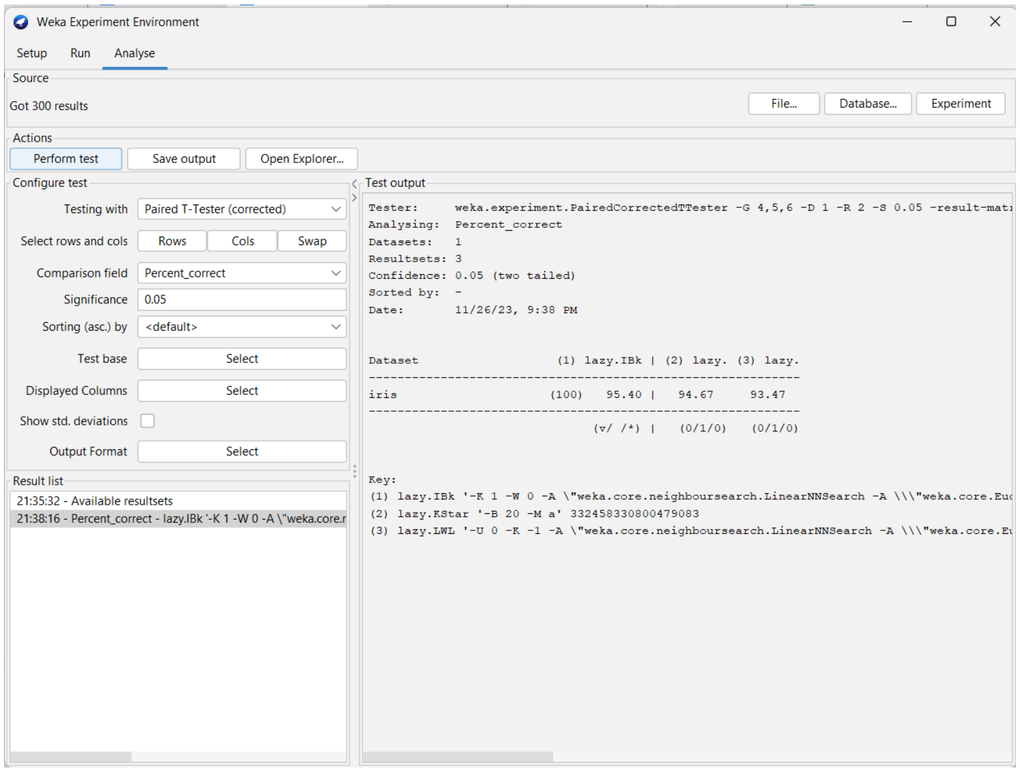
Show std. deviations: ☐

Output Format: Select

21:35:32 - Available resultsets

```
Available resultsets
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Euc
(2) lazy.KStar '-B 20 -M a' 332458330800479083
(3) lazy.LWL '-U 0 -K -1 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Eu
```

Under “Actions”, Click on “Perform Test”



Got 300 results

Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

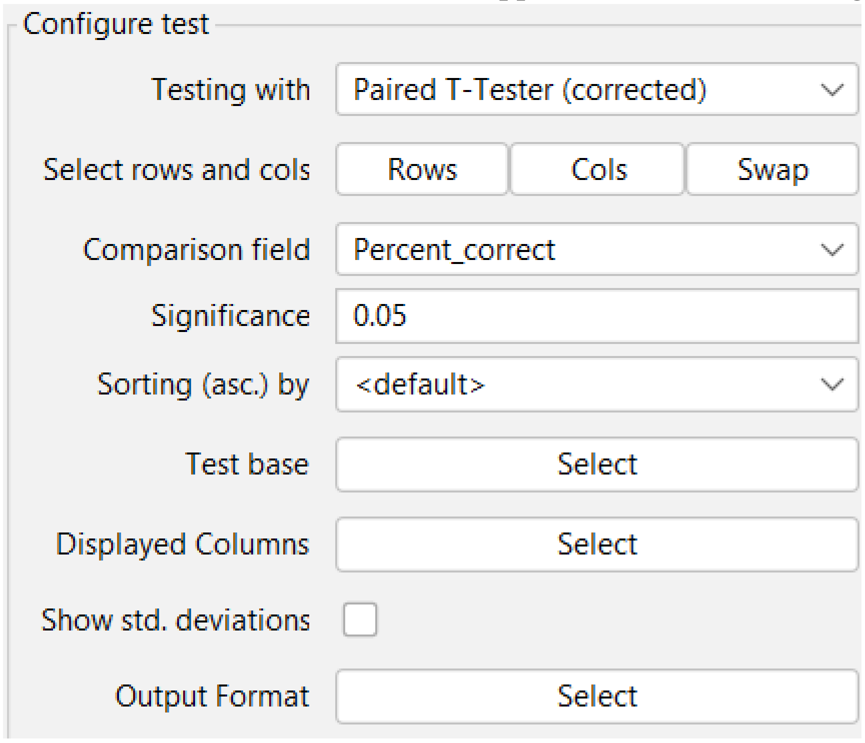
Sorting (asc.) by: <default>

Show std. deviations: ☐

21:35:32 - Available resultsets
21:38:16 - Percent_correct - lazy.IBk '-K 1 -W 0 -A \"weka.core.r

```
Tester:     weka.experiment.PairedCorrectedTTester -G 4,5,6 -D 1 -R 2 -S 0.05 -result-matr
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       11/26/23, 9:38 PM


Dataset                   (1) lazy.IBk | (2) lazy. (3) lazy.
------------------------------------------------------------
iris                     (100)   95.40 |   94.67     93.47
------------------------------------------------------------
                               (v/ /*) |   (0/1/0)   (0/1/0)


Key:
(1) lazy.IBk '-K 1 -W 0 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Euc
(2) lazy.KStar '-B 20 -M a' 332458330800479083
(3) lazy.LWL '-U 0 -K -1 -A \"weka.core.neighboursearch.LinearNNSearch -A \\\"weka.core.Eu
```

Here, Various configurations can be made such as the specific rows and columns in the dataset can be selected and also swapped as well as the sorting can be done.



Testing with: Paired T-Tester (corrected)

Comparison field: Percent_correct

Significance: 0.05

Sorting (asc.) by: <default>

Test base: Select

Displayed Columns: Select

Show std. deviations: ☐

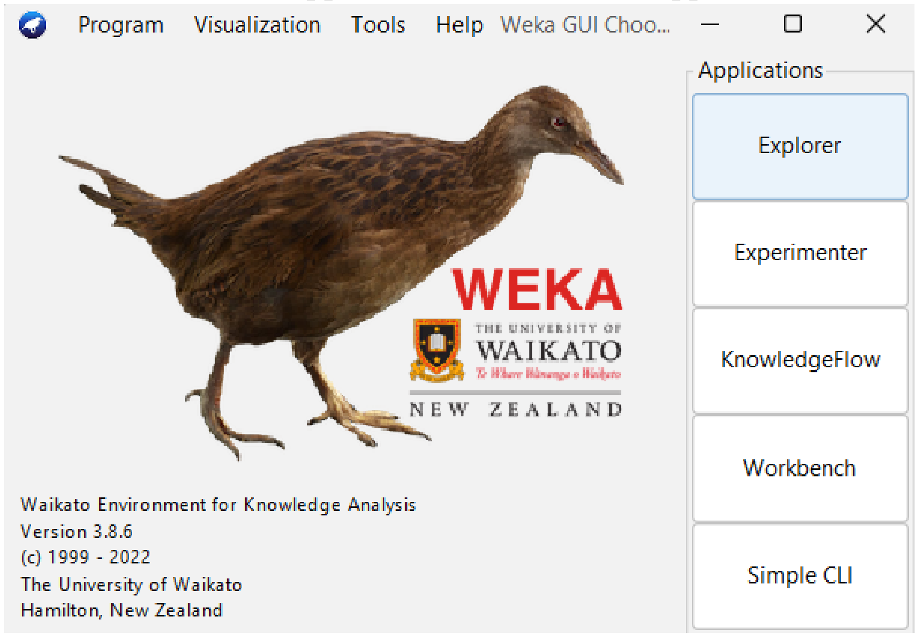
Output Format: Select

Practical 8

Implementation of KDD process in WEKA – Knowledge Flow

Step 1 :

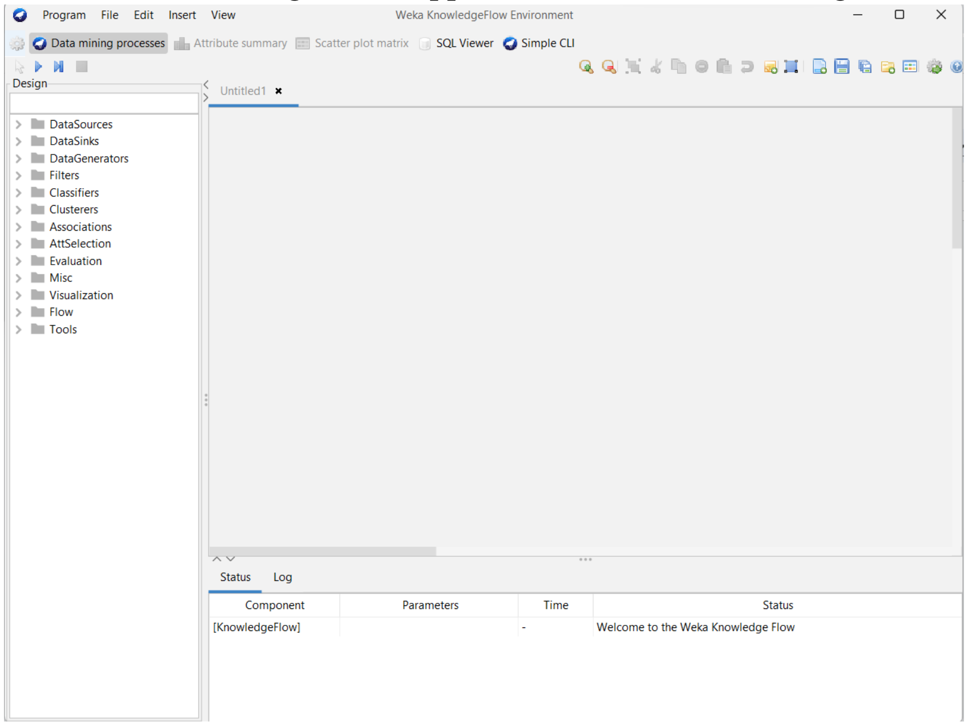
Open the WEKA application and the first page is displayed where you have the option to choose from various applications that WEKA supports.



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Step 2 :

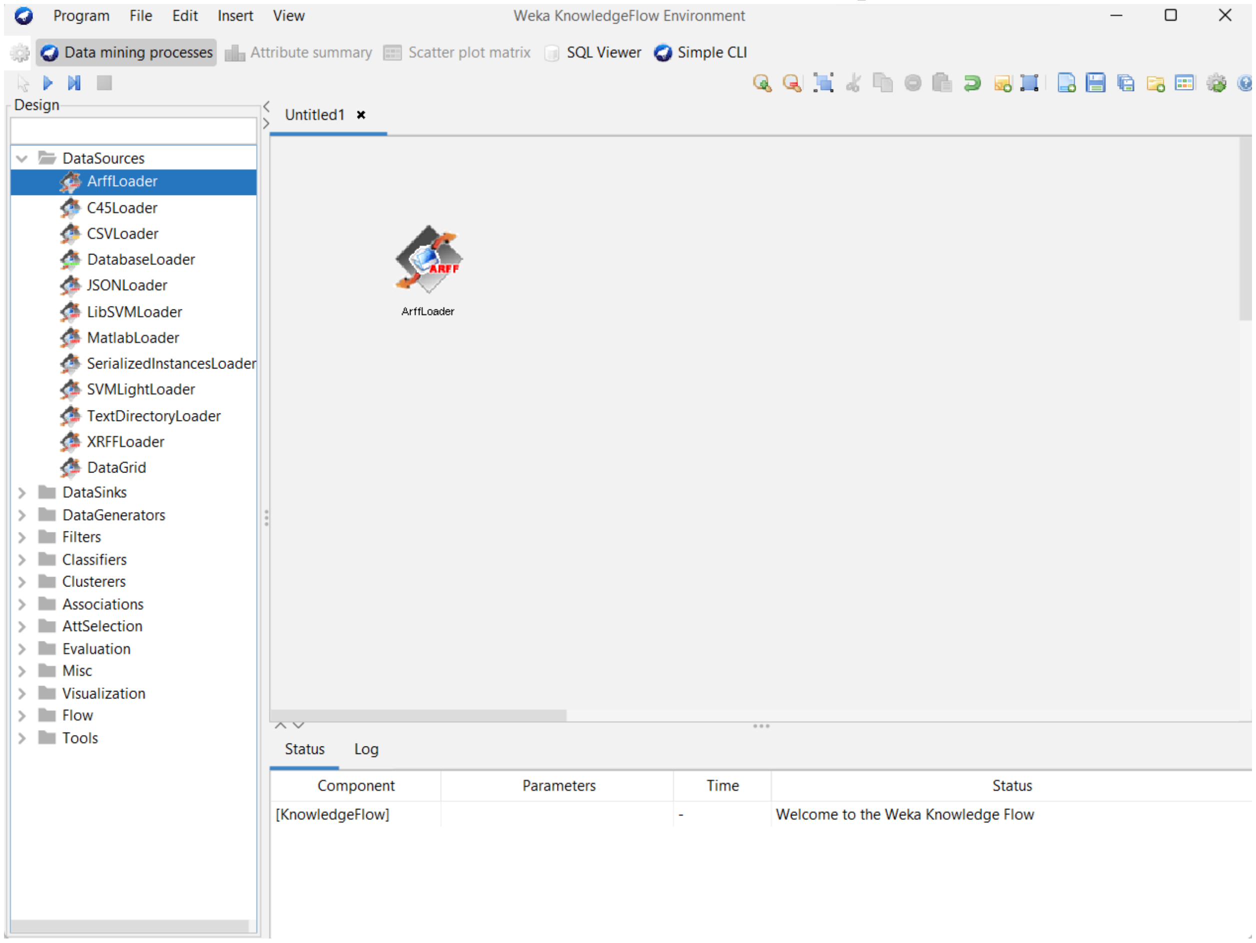
Select the "KnowledgeFlow" application and the below image is shown.



Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

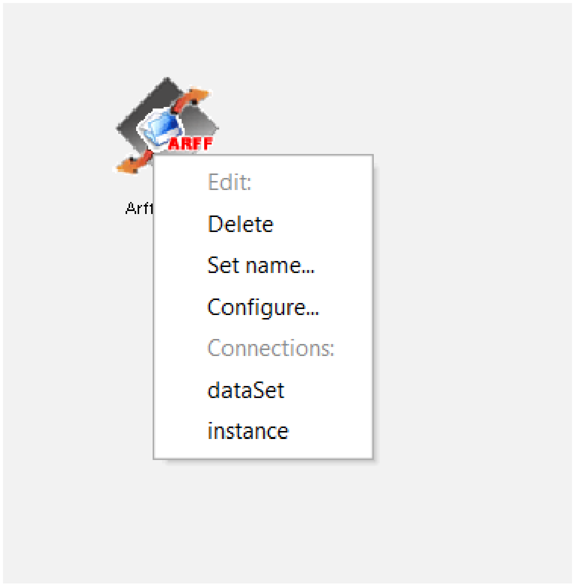
Step 3 :

Click on “DataSources” and then select “ArffLoader” and drop it on the screen.

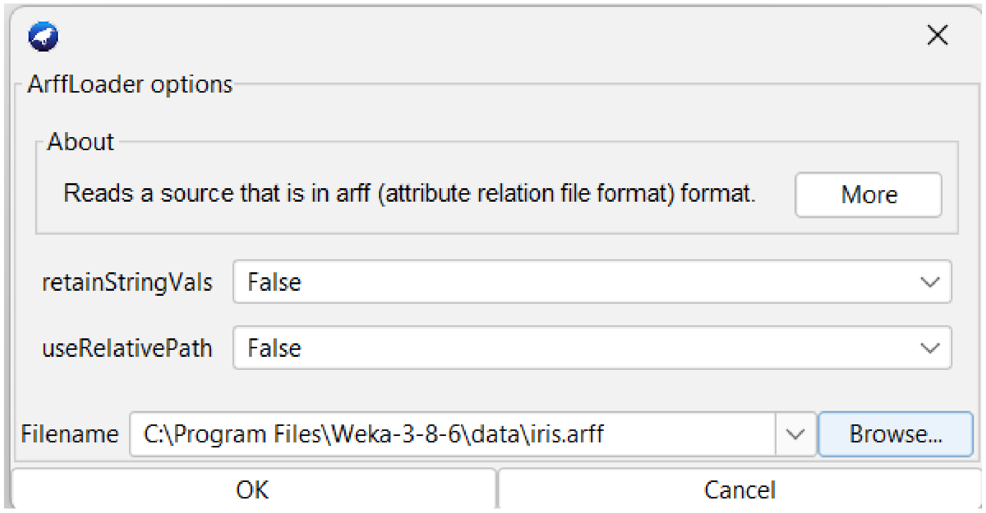


Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

Right click on “ArffLoader” on the screen and select “Configure”



"Browse" and choose the "iris" dataset. Then select "OK"



Reads a source that is in arff (attribute relation file format) format.

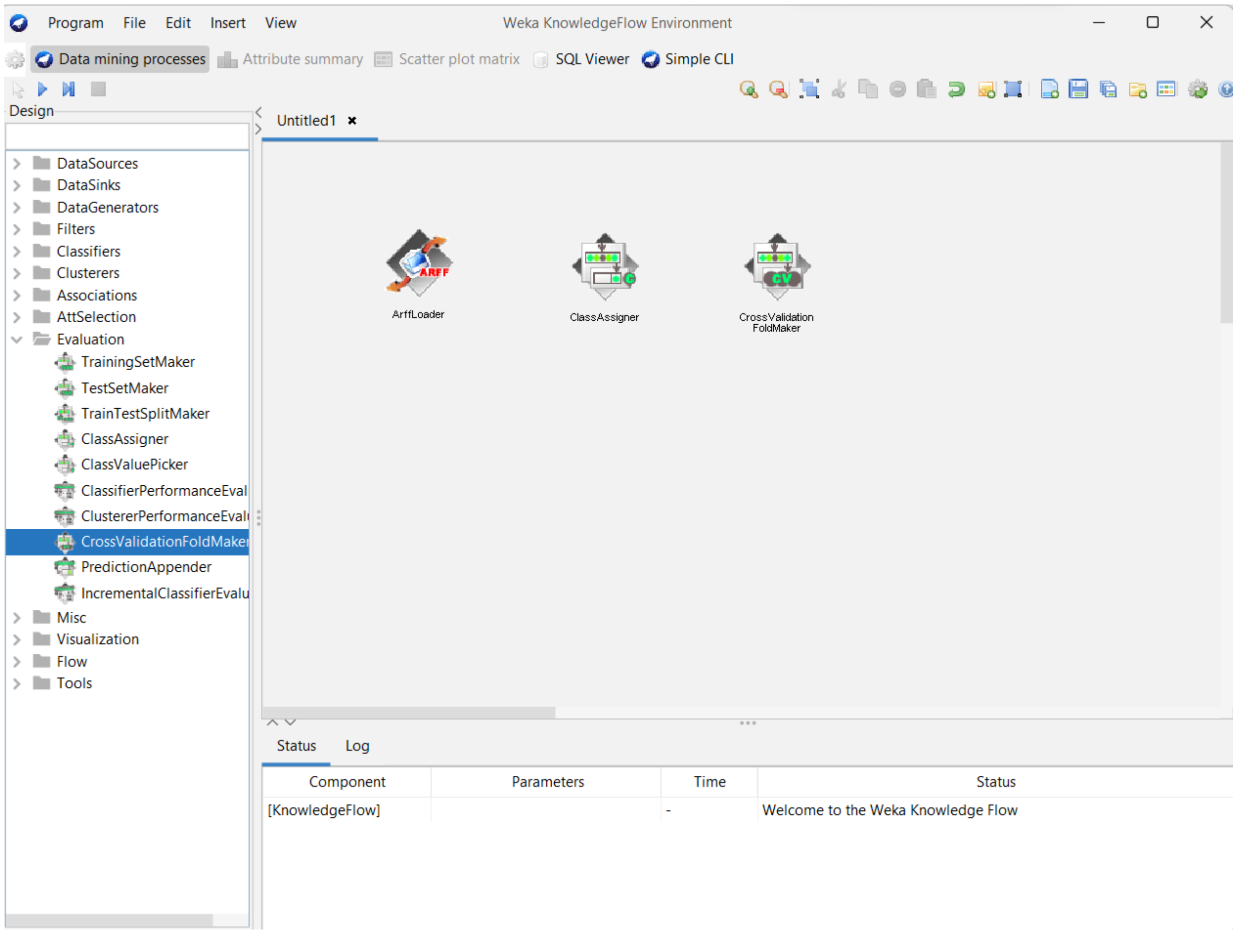
retainStringVals False

useRelativePath False

Filename C:\Program Files\Weka-3-8-6\data\iris.arff

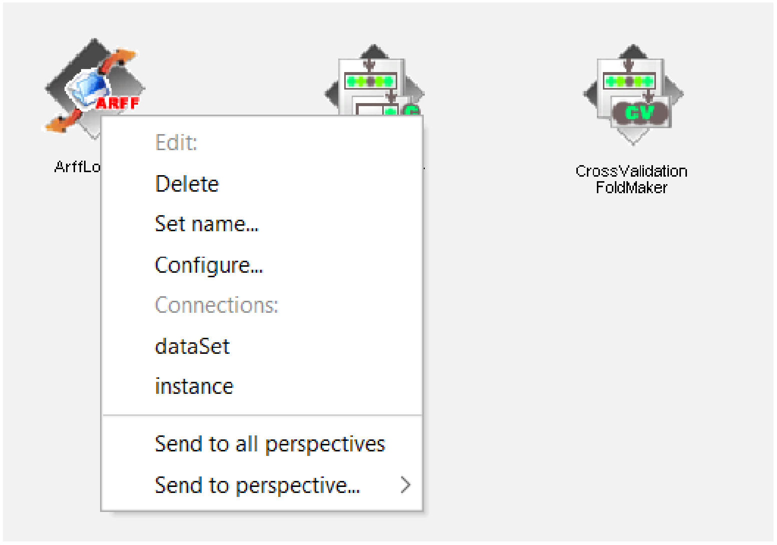
Step 4 :

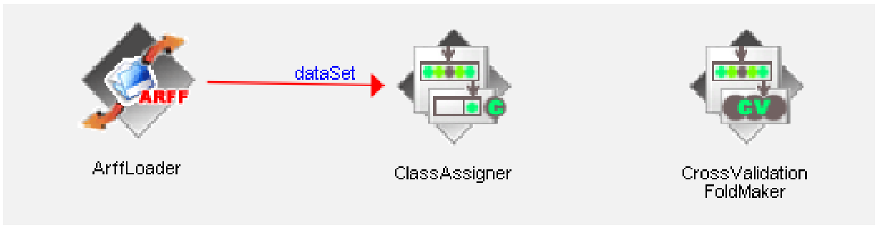
Click on "Evaluation" and select and drop "ClassAssigner" as well as "CrossValidationFoldMaker" onto the screen



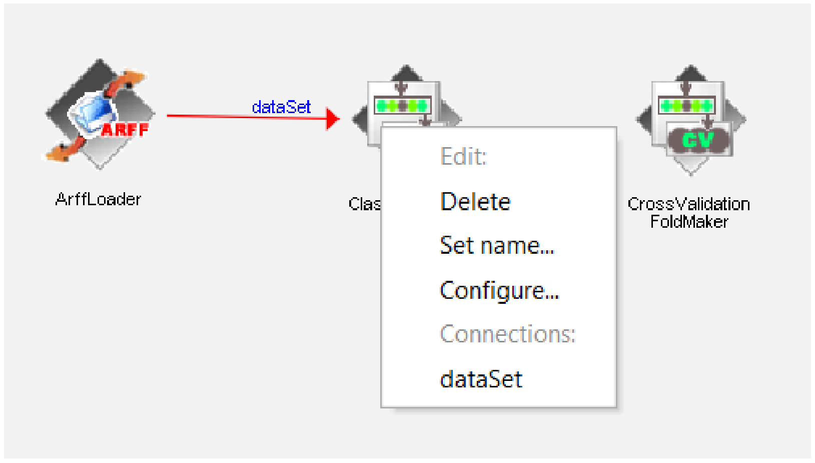
Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

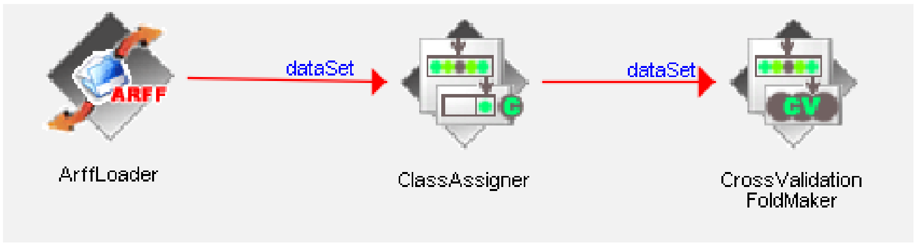
Right click on “ArffLoader” and click “dataSet”. Drag the arrow to “ClassAssigner”





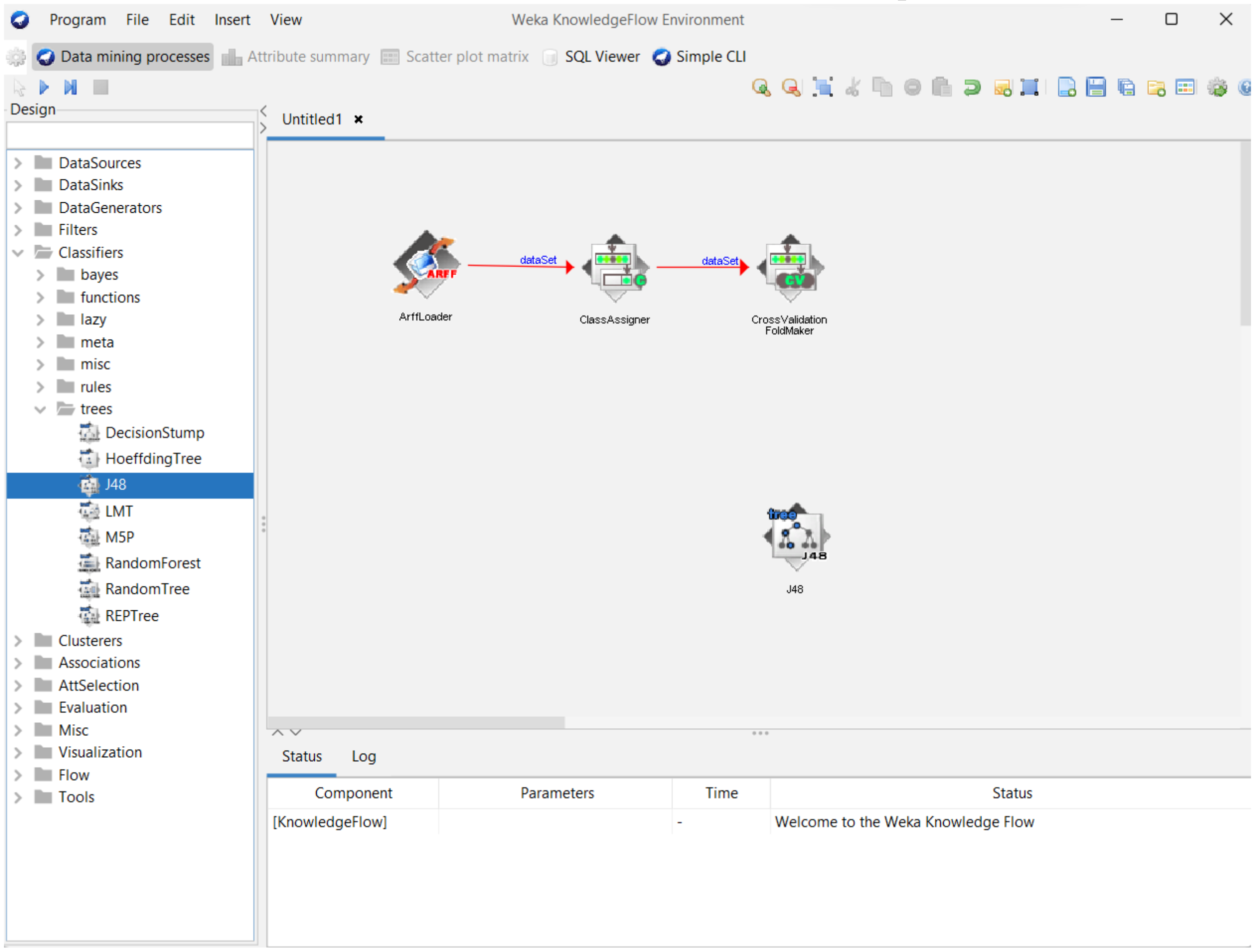
Right click on “ClassAssigner” and click “dataSet”. Drag the arrow to “CrossValidationFoldMaker”





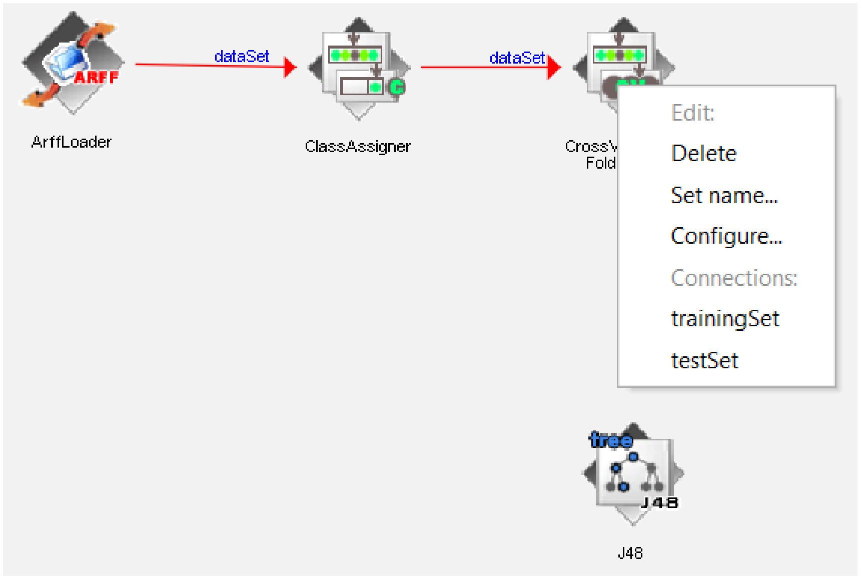
Step 5 :

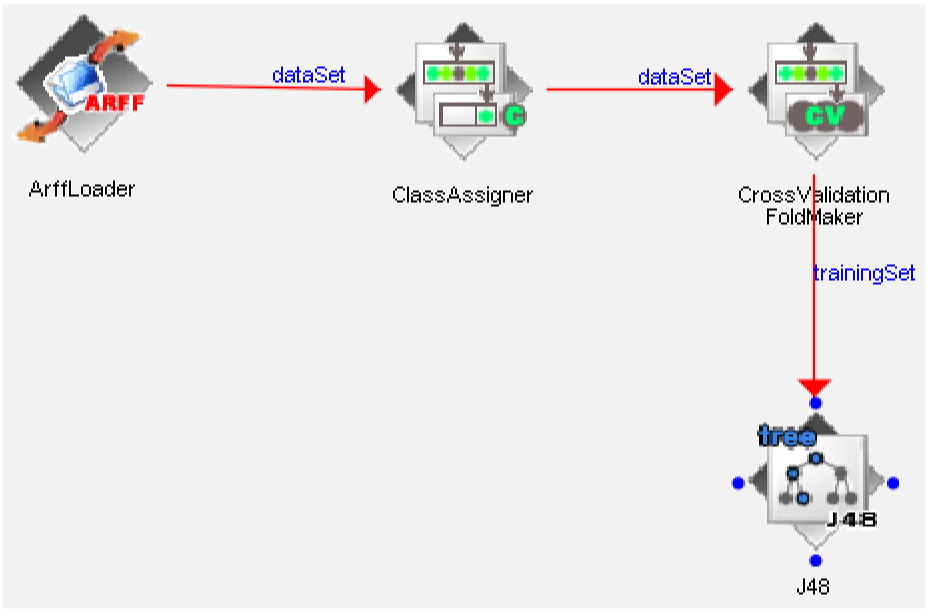
Click on “Classifiers” and select “trees”. Then, select “J48”. Drop it on the screen.

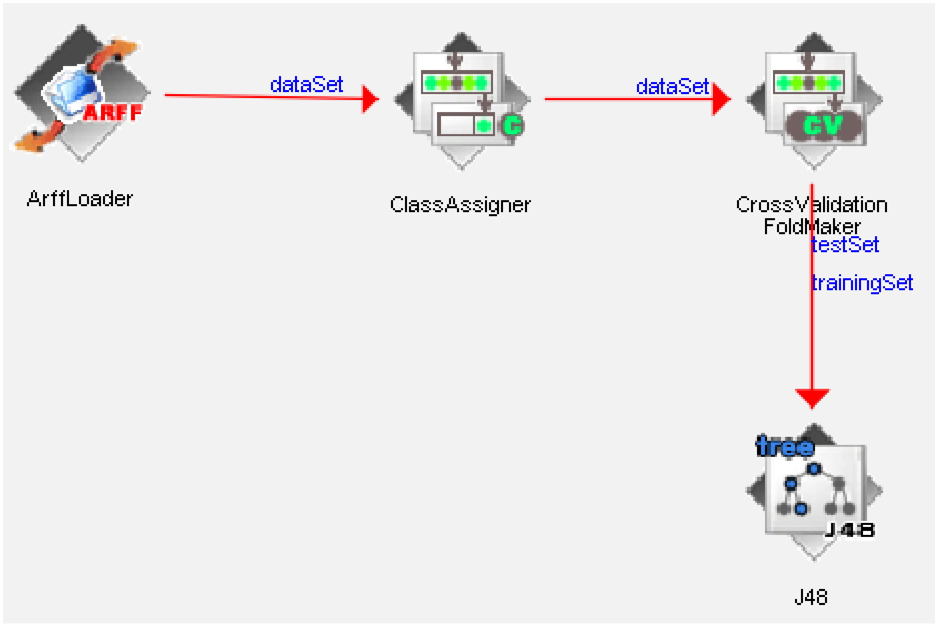


Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

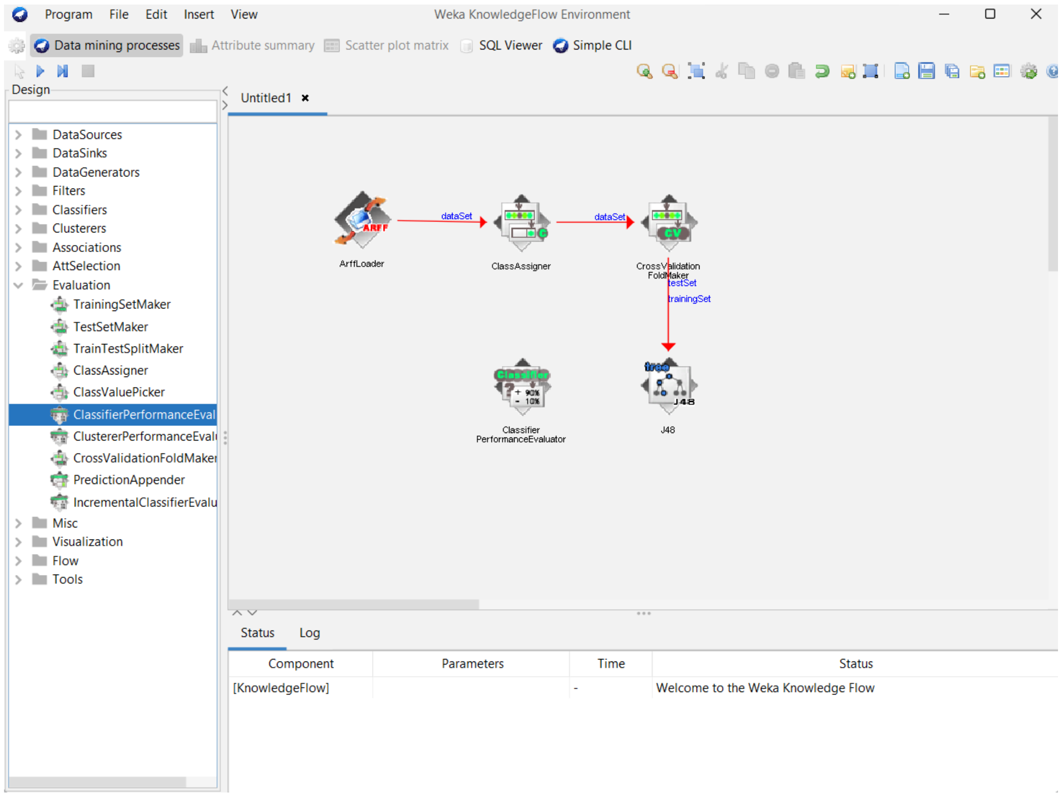
Right click on “CrossValidationFoldMaker” and click “testSet”. Drag the arrow to “J48”. Do the same with “trainingSet”.





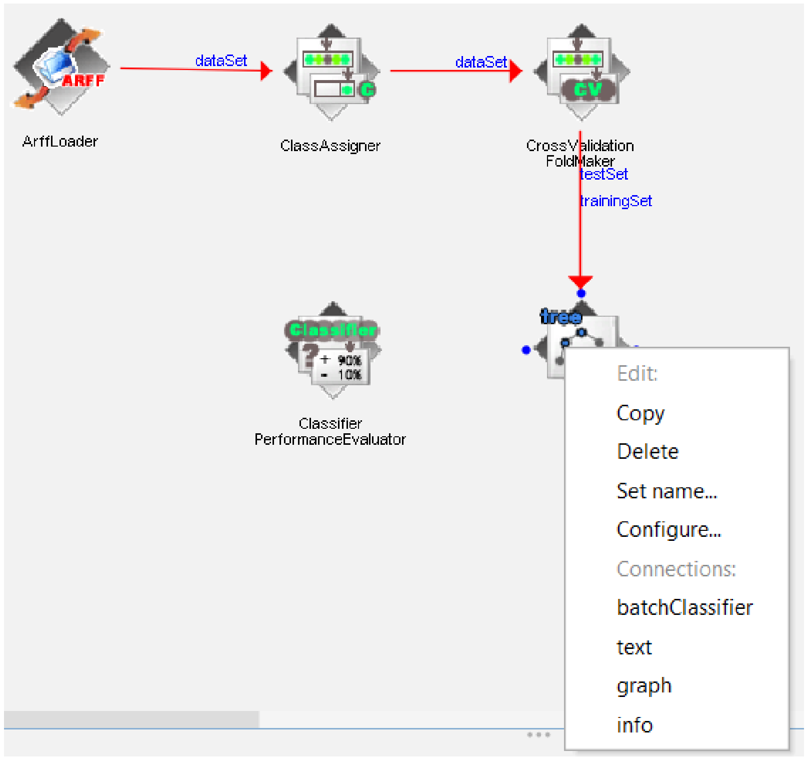


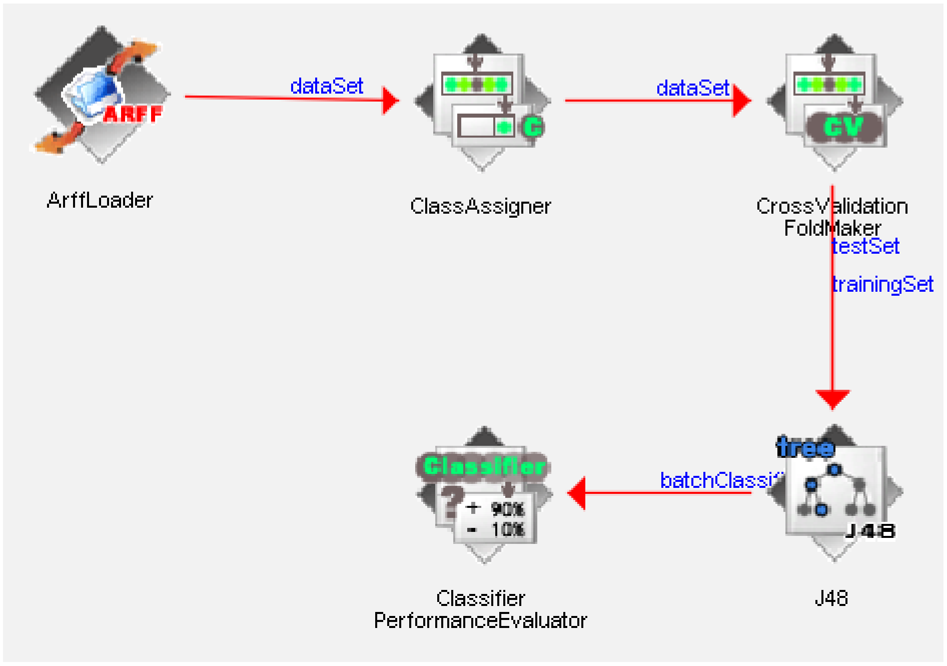
Step 6 :
Click on “Evaluation” and select and drop “ClassifierPerformanceEvaluator”onto the screen



Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

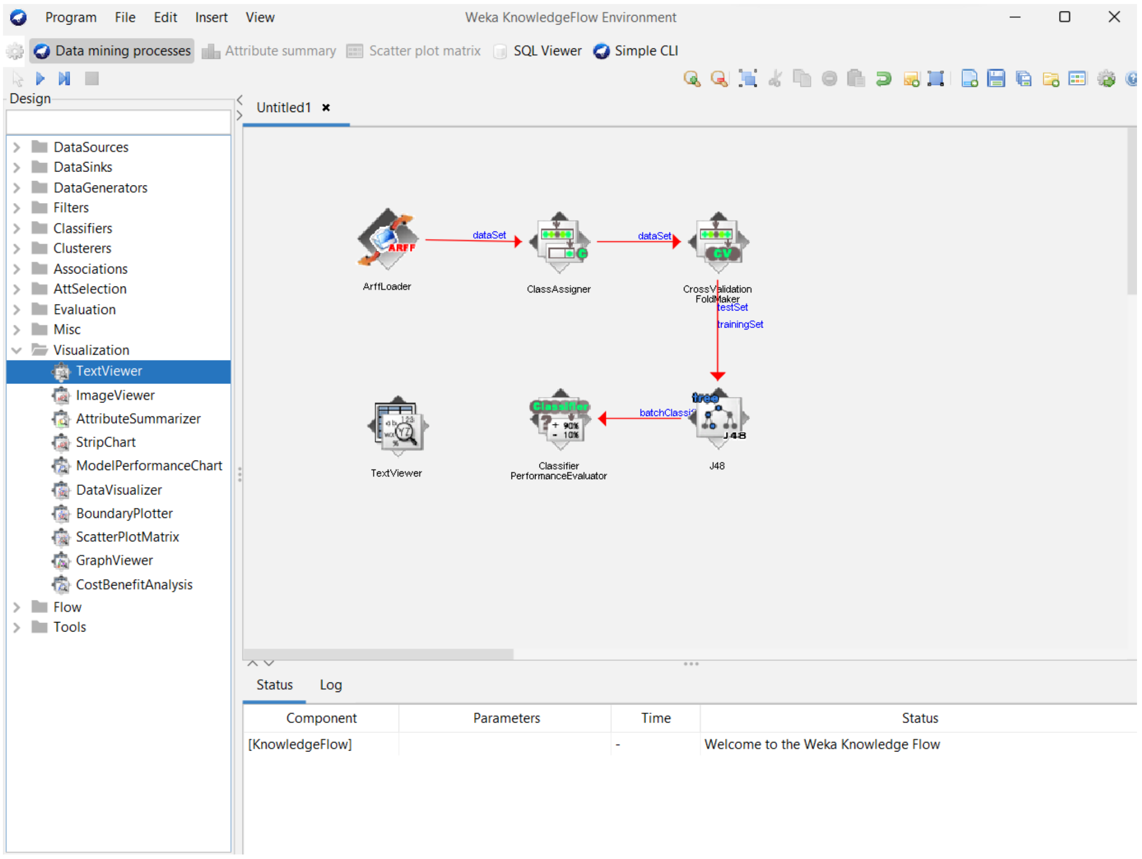
Right click on “J48” and select “batchClassifier”. Drag the arrow to “ClassifierPerformanceEvaluator”





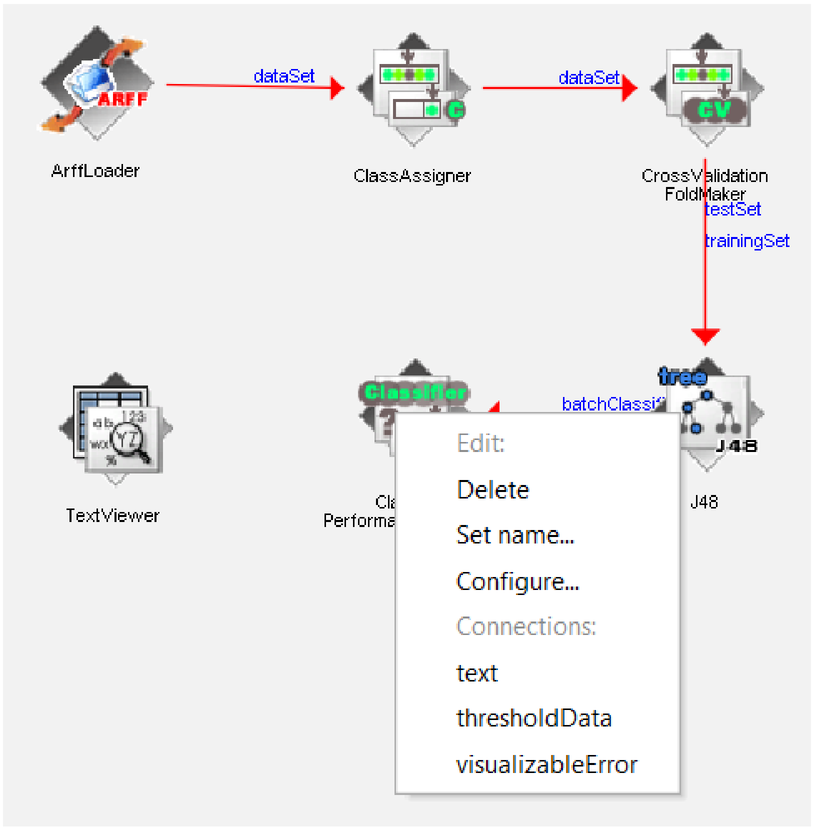
Step 7 :

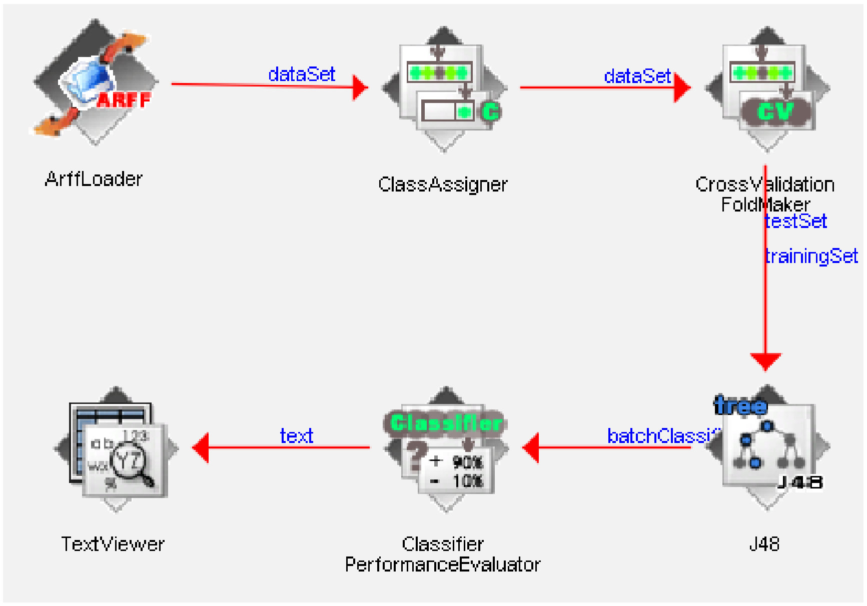
Click on "Visualization" and select and drop "TextViewer" onto the screen



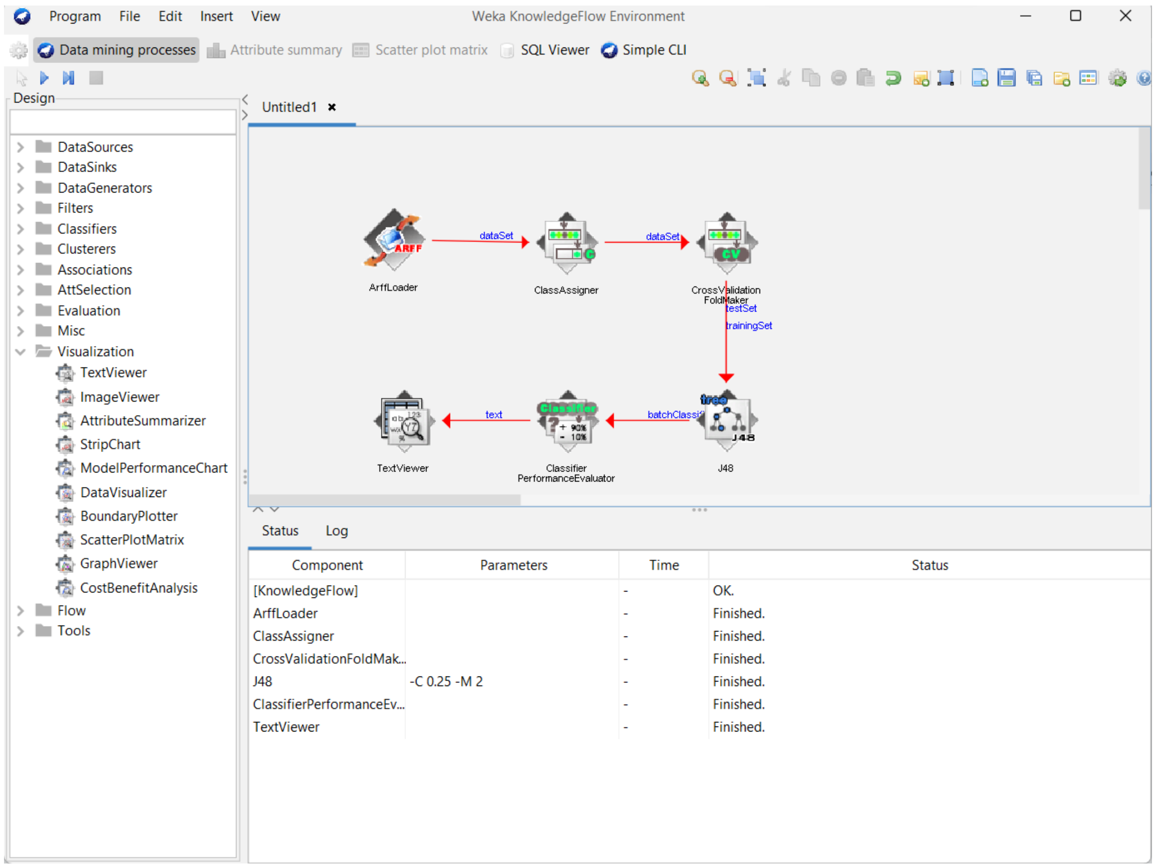
Component	Parameters	Time	Status
[KnowledgeFlow]		-	Welcome to the Weka Knowledge Flow

Right click on “ClassifierPerformanceEvaluator” and select “text”. Drag it to “TextViewer”





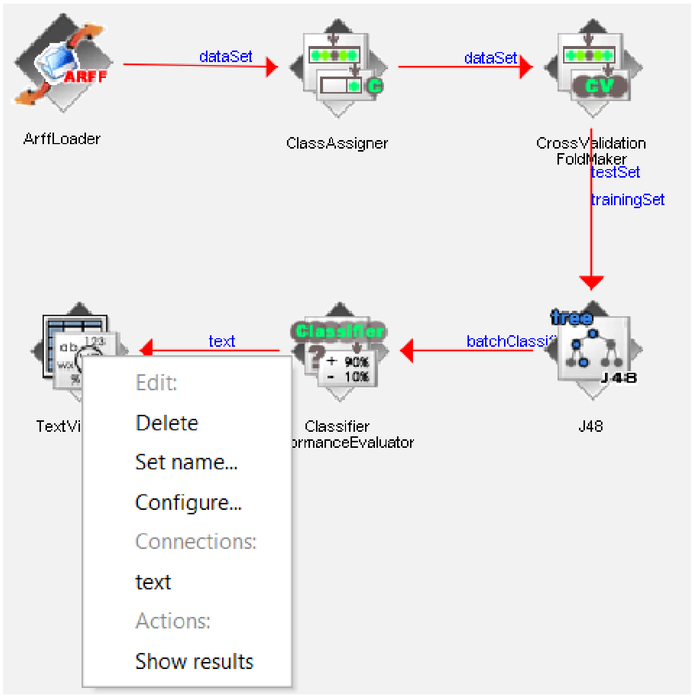
Step 8 :
Click on Run sign on the top left. The output is shown on “Status” below.

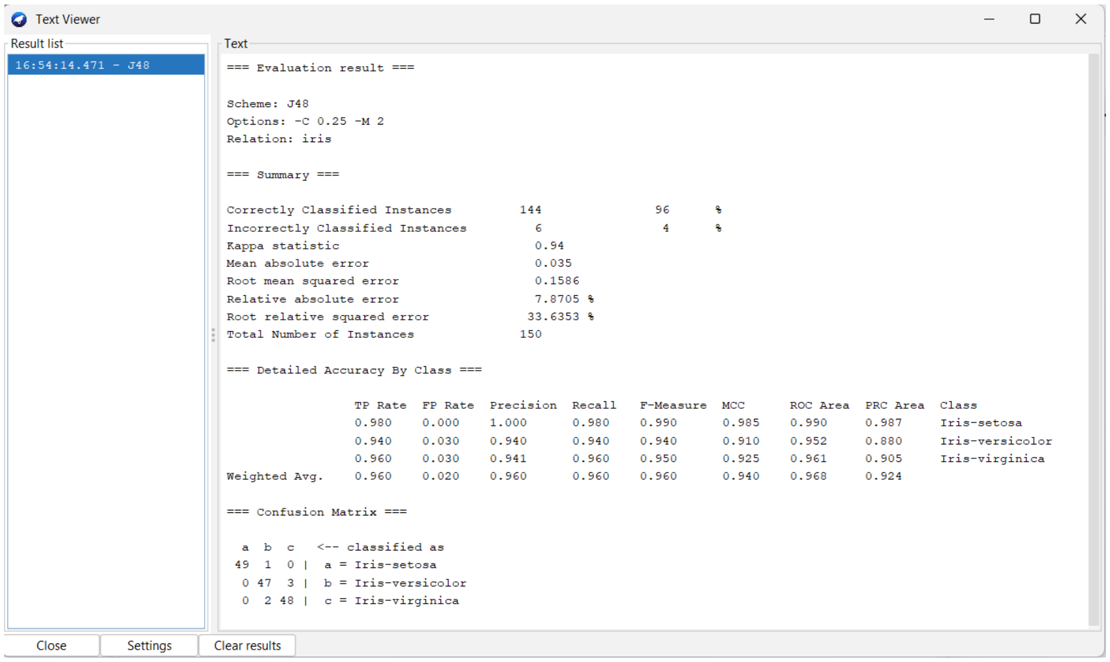


Component	Parameters	Time	Status
[KnowledgeFlow]		-	OK.
ArffLoader		-	Finished.
ClassAssigner		-	Finished.
CrossValidationFoldMak...		-	Finished.
J48	-C 0.25 -M 2	-	Finished.
ClassifierPerformanceEv...		-	Finished.
TextViewer		-	Finished.

Step 9 :

Right click on “TextViewer” and select “Show Results”





```
=== Evaluation result ===

Scheme: J48
Options: -C 0.25 -M 2
Relation: iris

=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-setosa
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-versicolor
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-virginica
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

For drawing diagram
datasources=arffloader(iris)
evaluation =crossvalidation and cross validation foldmaker(connect by dataset)

classifier=trees= J48(connect by training set and test set)
evaluation= ClassifierPerformanceEvaluator(connect with bath classifier)
Visualization= “TextViewer”(connect with text)

Practical 9

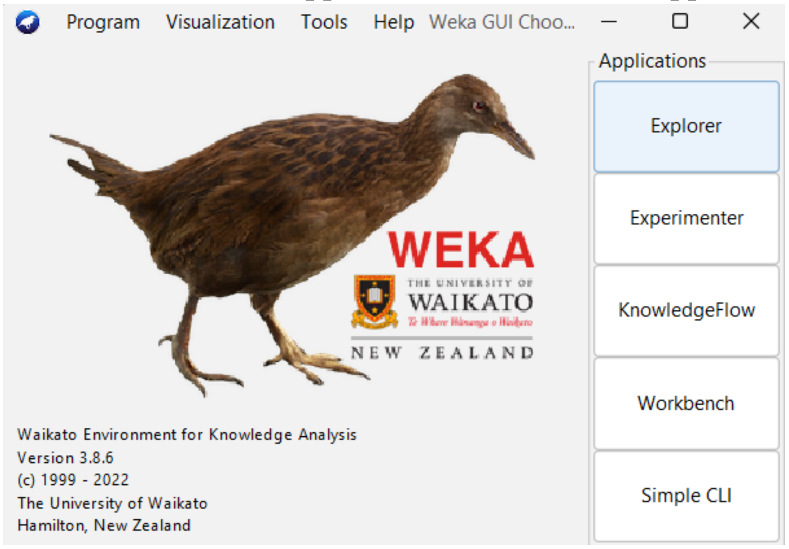
Practical on any Business Intelligence application.

a) Problem definition, identifying which data mining task is needed

b) Identify and use a standard data mining dataset available for the problem.

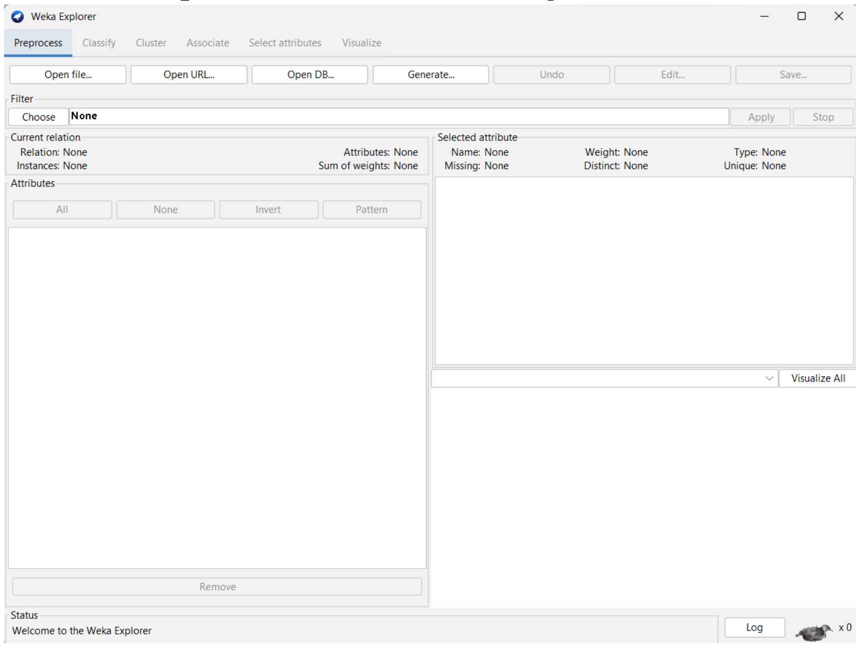
Problem Definition : Classify Iris flowers into species based on features.
Data Mining Task : Classification.

Step 1 :
Open the WEKA application and the first page is displayed where you have the option to choose from various applications that WEKA supports.



Waikato Environment for Knowledge Analysis
Version 3.8.6
(c) 1999 - 2022
The University of Waikato
Hamilton, New Zealand

Step 2 :
Click on “Explorer” and the below image is shown



Filter: None

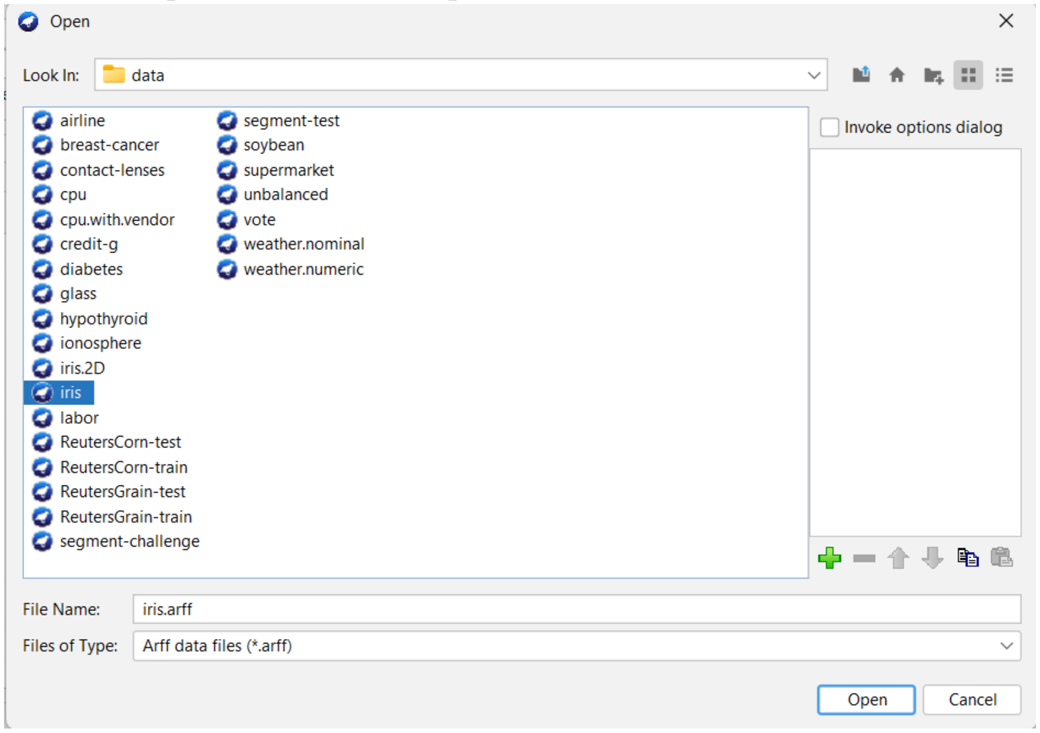
Relation: None
Instances: None
Attributes: None
Sum of weights: None

Name: None
Missing: None
Weight: None
Distinct: None
Type: None
Unique: None

Welcome to the Weka Explorer

Step 3 :

Click on "Open File" and then open the "iris" dataset.

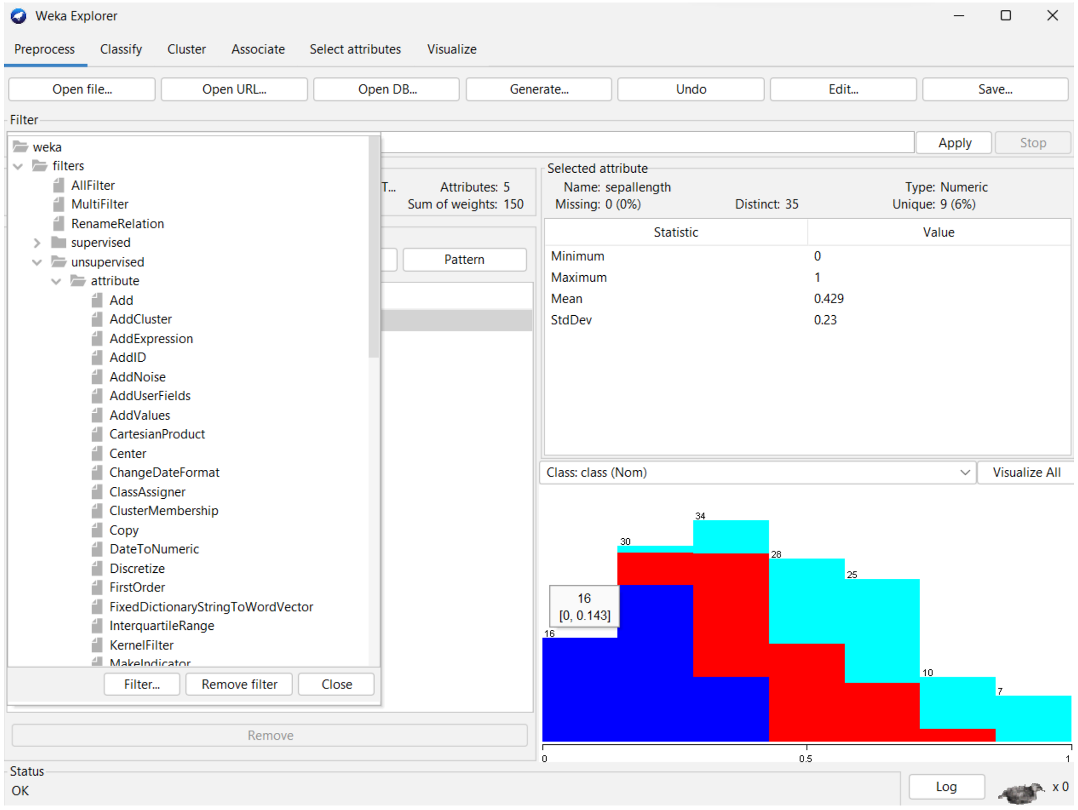


File Name: iris.arff

Files of Type: Arff data files (*.arff)

Step 4 :

Click on the "Choose" button. Choose the "Normalize" filter and then click "Apply"



Attributes: 5
Sum of weights: 150

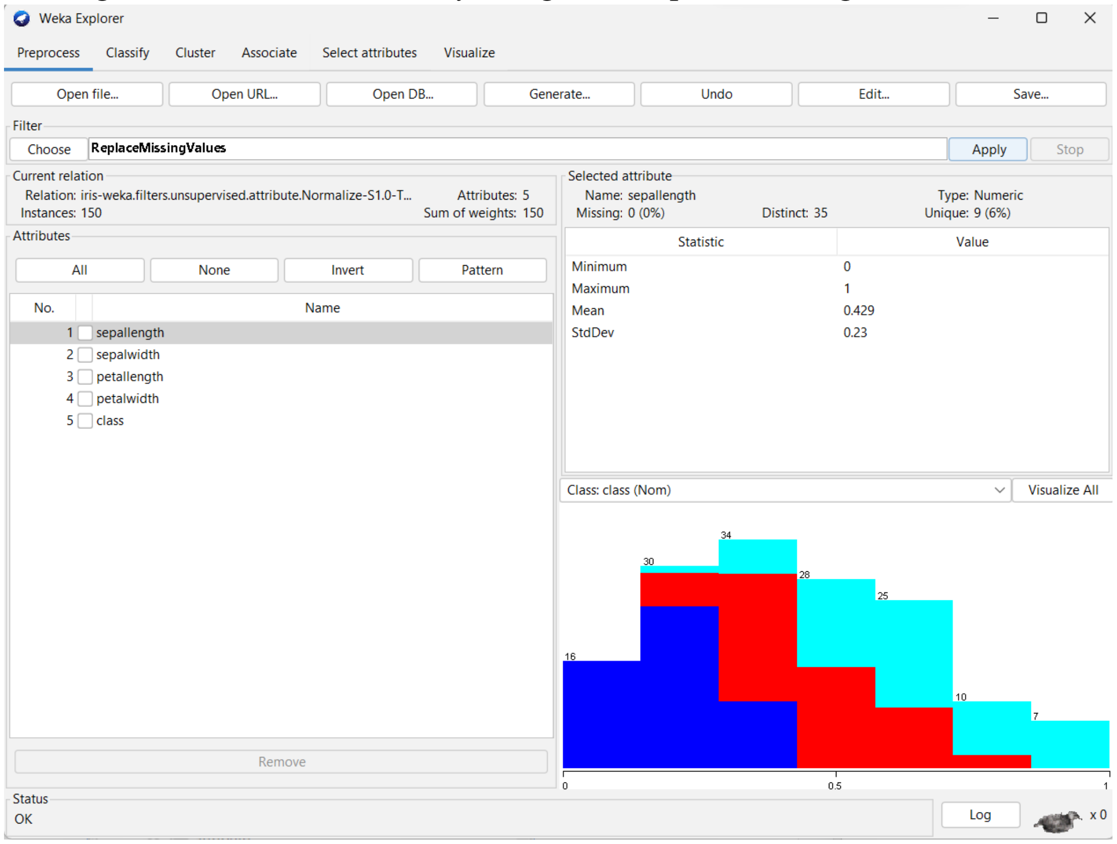
Selected attribute
Name: sepallength
Type: Numeric
Missing: 0 (0%)
Distinct: 35
Unique: 9 (6%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.429
StdDev	0.23

Class: class (Nom)

Status
OK

Missing values can be checked by using the "ReplaceMissingValues" filter.



Filter: ReplaceMissingValues

Current relation
Relation: iris-weka.filters.unsupervised.attribute.Normalize-S1.0-T... Attributes: 5
Instances: 150 Sum of weights: 150

Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class

Selected attribute
Name: sepallength Type: Numeric
Missing: 0 (0%) Distinct: 35 Unique: 9 (6%)

Statistic	Value
Minimum	0
Maximum	1
Mean	0.429
StdDev	0.23

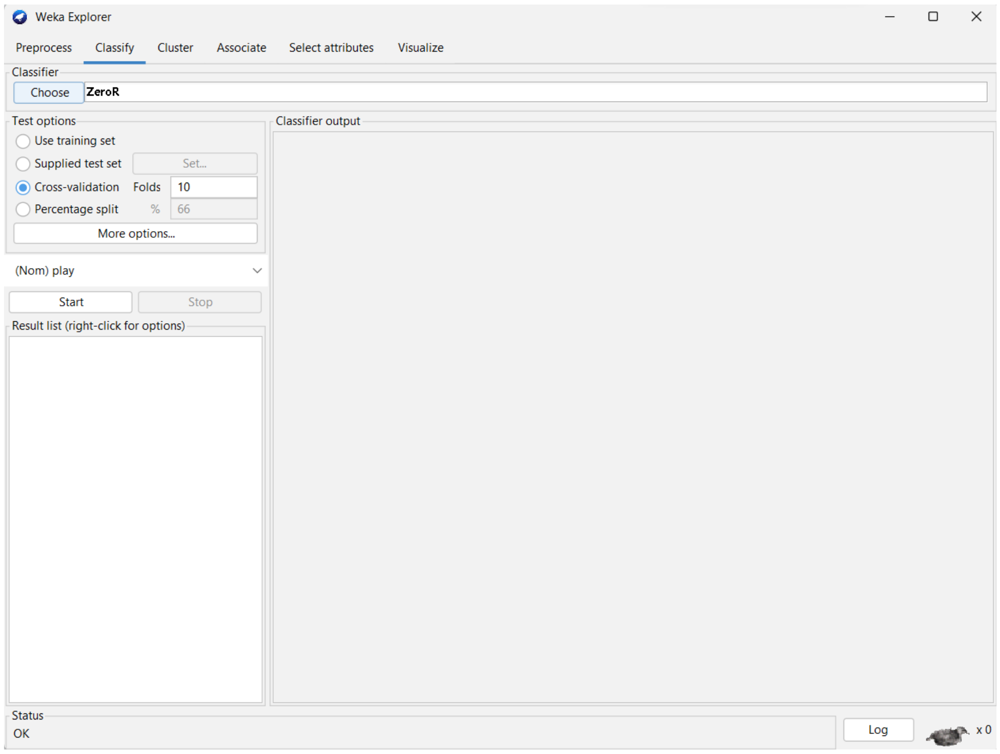
Class: class (Nom)

Status
OK

x 0

Step 5 :

Go to the "Classify" tab



Classifier: ZeroR

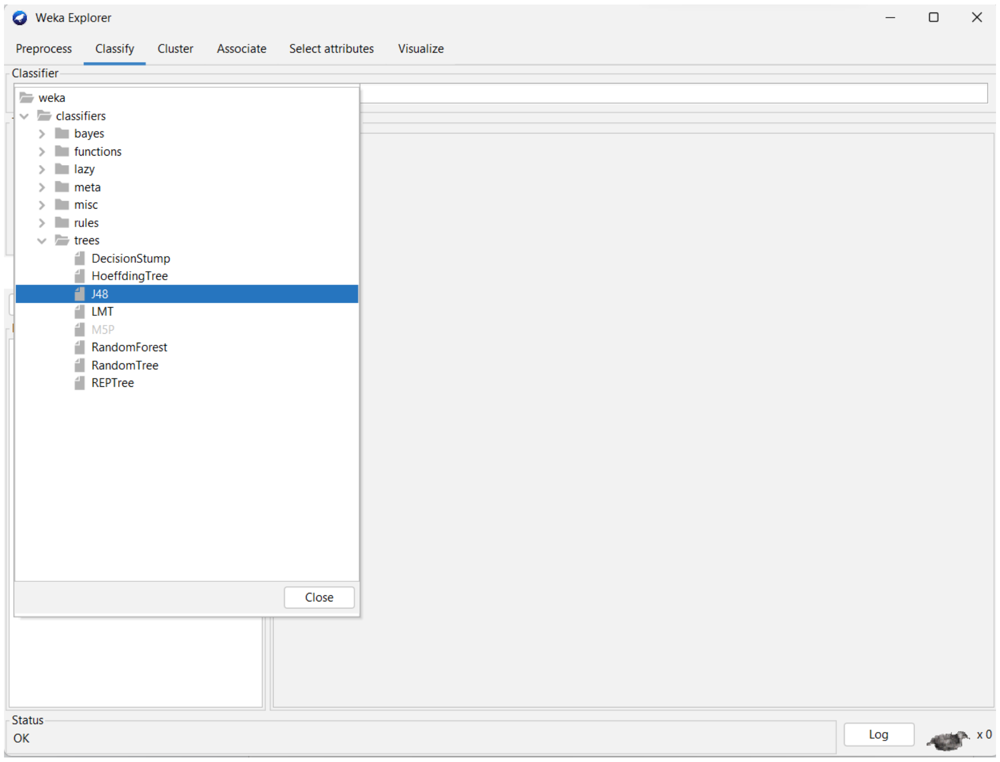
Test options
Use training set
Supplied test set
Cross-validation Folds 10
Percentage split % 66

(Nom) play

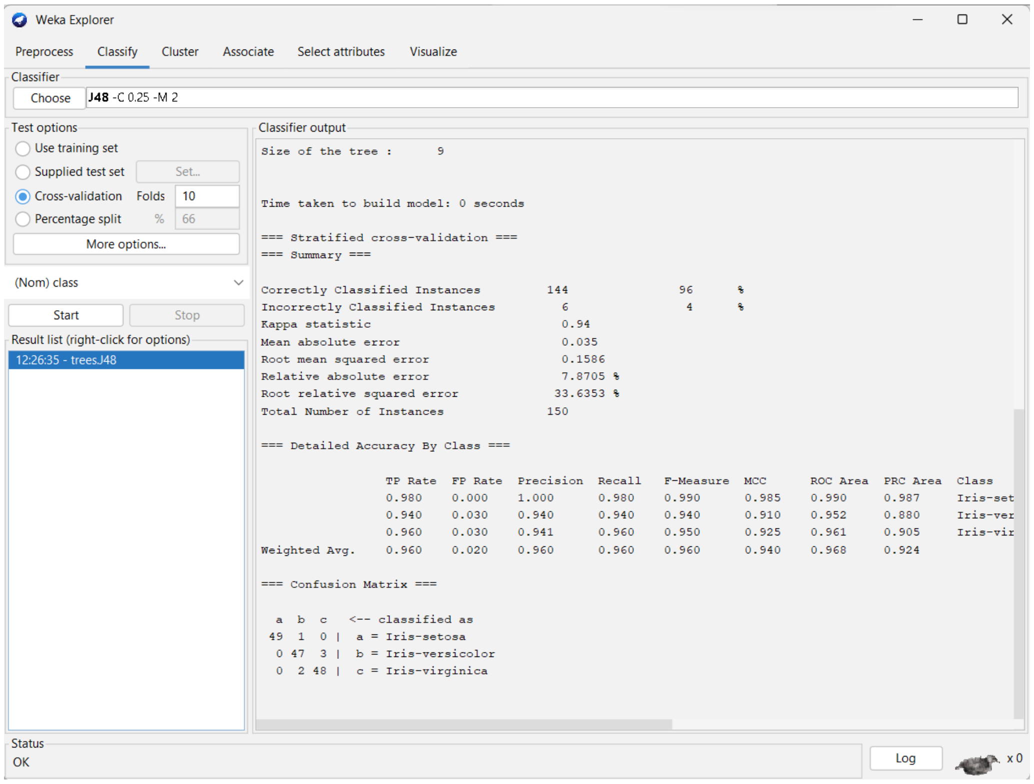
Status
OK

x 0

Click on "Choose" then select "J48" from "trees".



Select the column required, then click on "Start". Here. the column chosen is "class"



Classifier: J48 -C 0.25 -M 2

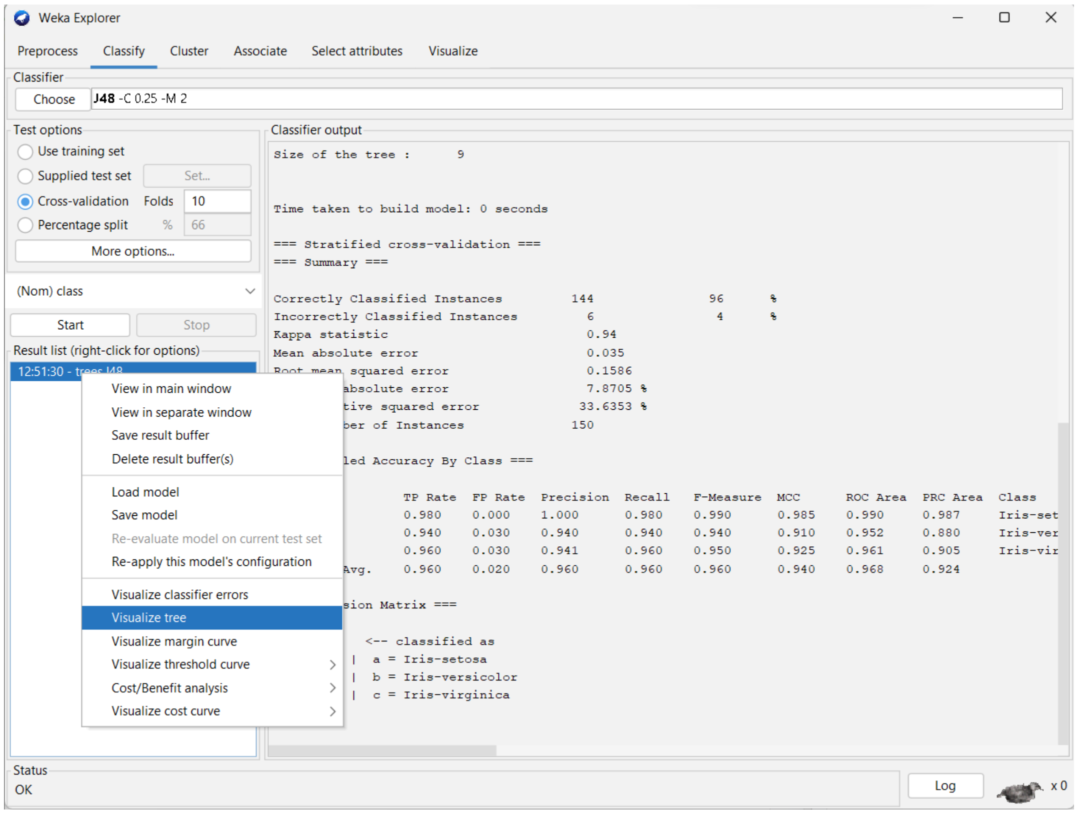
Test options: Cross-validation, Folds 10; Percentage split % 66

(Nom) class

12:26:35 - trees.J48

```
Size of the tree :      9


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
Relative absolute error                  7.8705 %
Root relative squared error             33.6353 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-set
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-ver
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-vir
Weighted Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

=== Confusion Matrix ===

  a  b  c   <-- classified as
 49  1  0 |  a = Iris-setosa
  0 47  3 |  b = Iris-versicolor
  0  2 48 |  c = Iris-virginica
```

Right click on the J48 tab and select “Visualize tree”



Classifier: J48 -C 0.25 -M 2

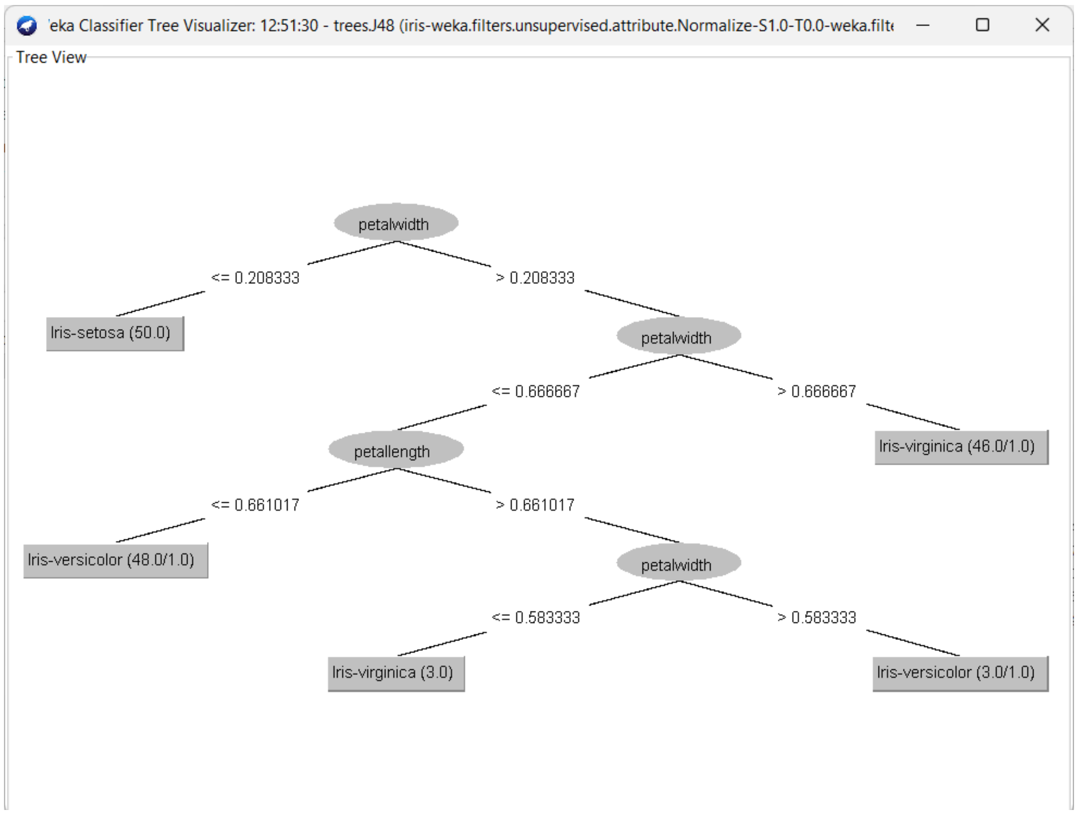
Test options: Use training set / Supplied test set / [✓ Cross-validation] Folds 10 / Percentage split % 66

(Nom) class

Result list: 12:51:30 - trees.J48

Classifier output

```
Size of the tree :      9


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         144               96      %
Incorrectly Classified Instances         6                4      %
Kappa statistic                          0.94
Mean absolute error                      0.035
Root mean squared error                  0.1586
         absolute error                  7.8705 %
         tive squared error             33.6353 %
         ber of Instances              150

         led Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.980    0.000    1.000      0.980    0.990      0.985    0.990     0.987     Iris-set
                 0.940    0.030    0.940      0.940    0.940      0.910    0.952     0.880     Iris-ver
                 0.960    0.030    0.941      0.960    0.950      0.925    0.961     0.905     Iris-vir
         Avg.    0.960    0.020    0.960      0.960    0.960      0.940    0.968     0.924

         sion Matrix ===

            <-- classified as
          |   a = Iris-setosa
          |   b = Iris-versicolor
          |   c = Iris-virginica
```


'eka Classifier Tree Visualizer: 12:51:30 - trees.J48 (iris-weka.filters.unsupervised.attribute.Normalize-S1.0-T0.0-weka.filt
Tree View
petalwidth
<= 0.208333
> 0.208333
Iris-setosa (50.0)
petalwidth
<= 0.666667
> 0.666667
petallength
Iris-virginica (46.0/1.0)
<= 0.661017
> 0.661017
Iris-versicolor (48.0/1.0)
petalwidth
<= 0.583333
> 0.583333
Iris-virginica (3.0)
Iris-versicolor (3.0/1.0)

Now, the model can also be used to experiment with different classifiers and other algorithms to enhance their performance.
It can then be interpreted and used to make predictions on new instances of iris flowers and can also be deployed for classification of iris flowers in real-time.