Experiment 1

Shashwat Shah
60004220126
Div B C2-2

Aim: Perform data preprocessing task using weka data mining tool.

Theory: weka is an opensource software that performs tools for pre-processing implementation of several machine learning algorithms & visual tools.

We can use the weka tool to perform

1) Pre-processing
2) Classification
3) Clustering
4) Association Rule
5) Visualization.

Pre-processing - It involves cleaning & transforming raw data into a format suitable for analysis. The goal is to enhance the data & ensure that it is well suited for a specific requirement of data mining task.

It involves steps such as:-

1. Data cleaning - Addressing errors & inconsistencies in data
2. Data integration - Combine data from different sources.
3. Data transformation - changing the format or structure of the data.
4. Data reduction - Reducing the volume but producing the same result.
5. Data Discretization - Transforming continuous data into discrete categories

(6) Handling noisy data - Make sure of the data.
(7) Normalization - scaling numerical features to the range
(8) Handling the imbalanced data.
(9) Data Imputation. - Filling the missing data.
(10) Feature Engineering

Conclusion: Thus the weka mining tool was explored using the weka tool tasks like pre-processing.

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Name: Shashwat Shah

SAP-ID: 60004220126

TY BTECH DIV B, Batch : C22

LAB EXPERIMENT NO. 01

Aim: Perform data Pre-processing task using Weka data mining tool

Theory:

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

Tasks performed through Weka:

Preprocessing:

Classification:

Clustering:

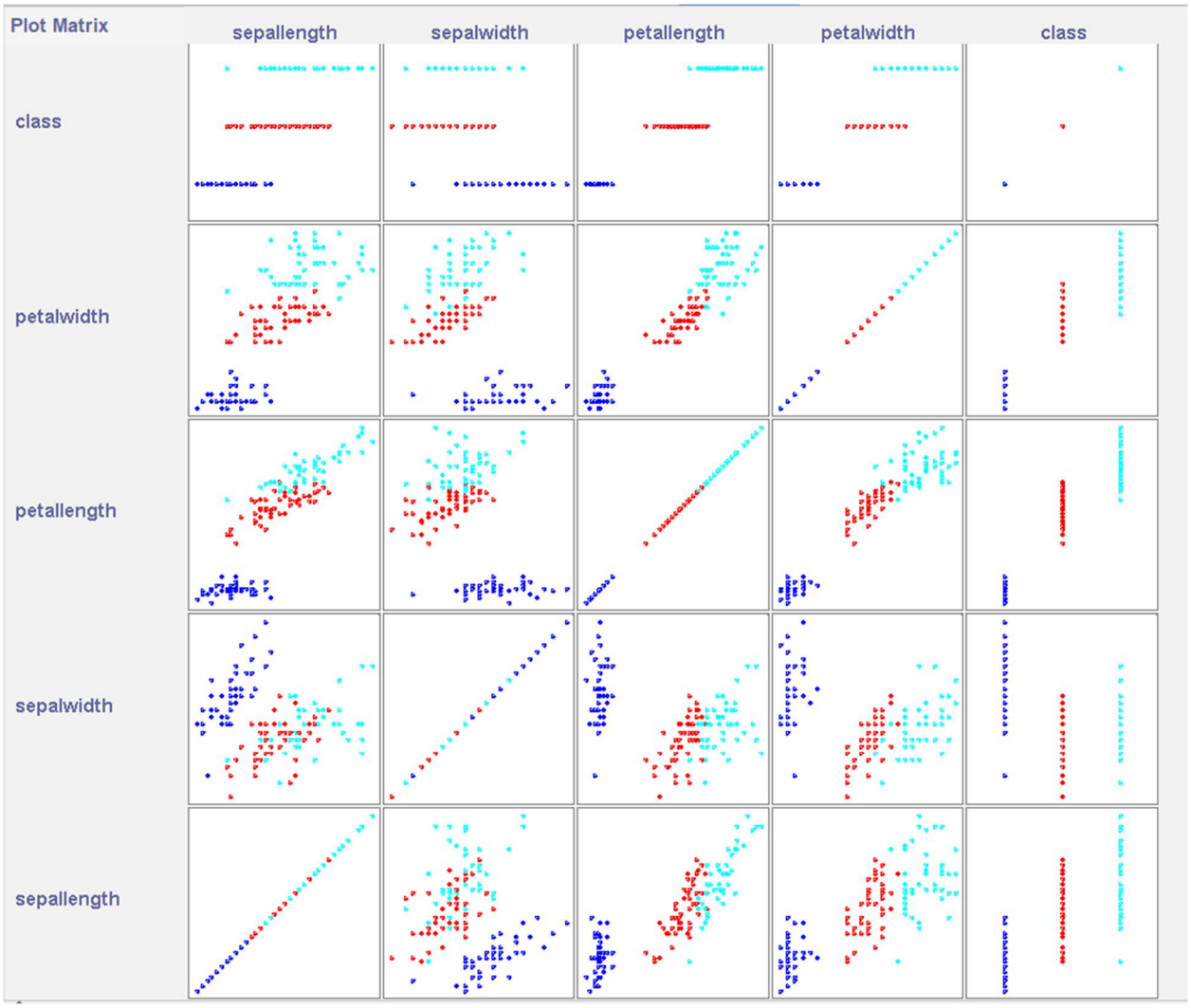
Association Rule:

Select Attributes:

Visualization:

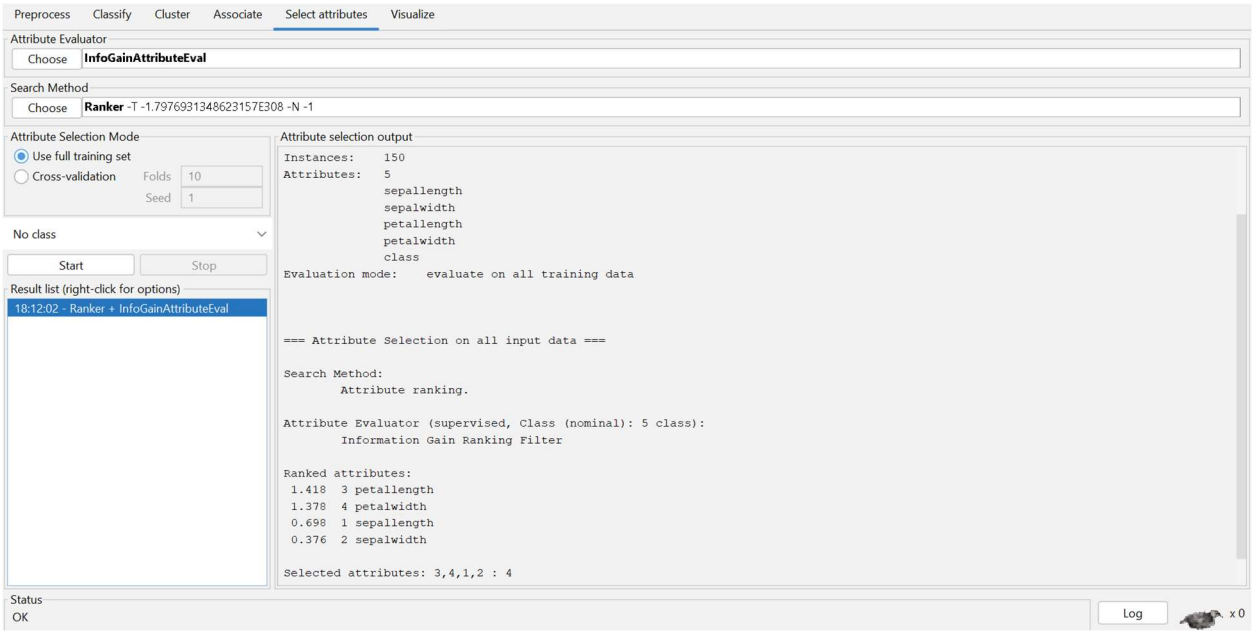
Preprocessing activities to be observed in Weka:

1. **Visualization:** Visualize scatter plot for all the attributes from dataset selected from Weka. Determine correlation if any using these plots for different datasets



Upon observing the scatter plot in the visualize section, we can observe certain correlations within
the attributes. Some of them have been listed below
a. Petal length vs Sepal length: Positive correlation
b. Petal length vs Petal width: Positive correlation
c. Petal width vs Sepa length: Positive correlation

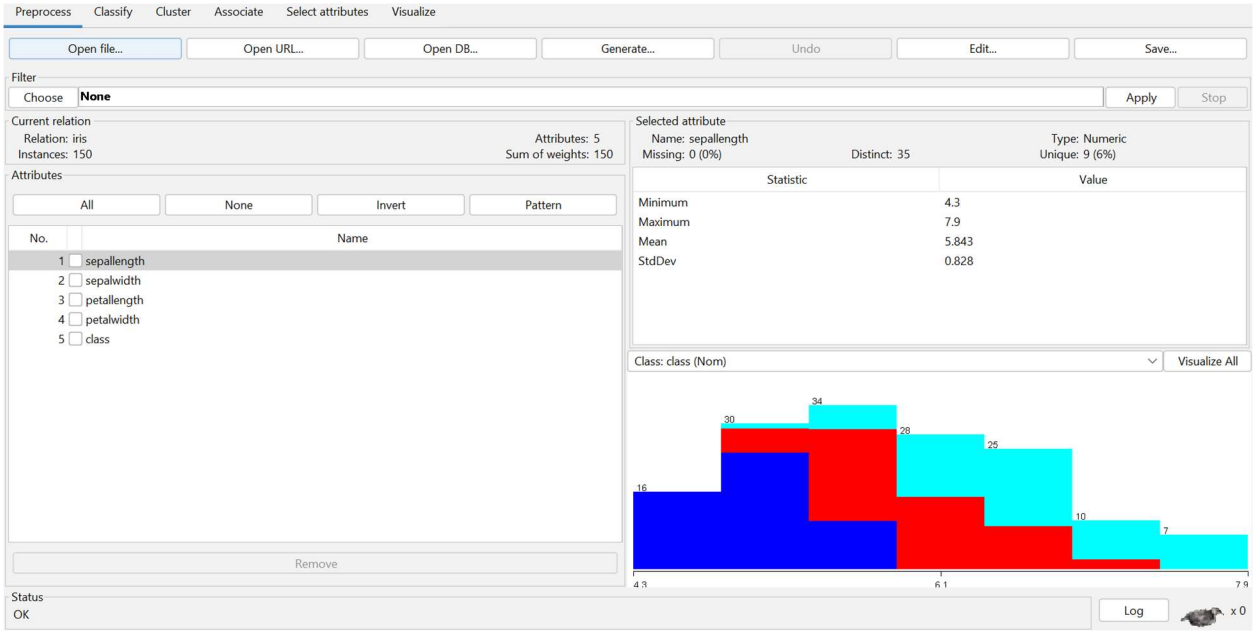
2. **Select Attributes:** Apply suitable feature selection filter like GainRatio etc to choose relevant attributes from the list of attributes. Observe the ranks / priority provided by the filter.

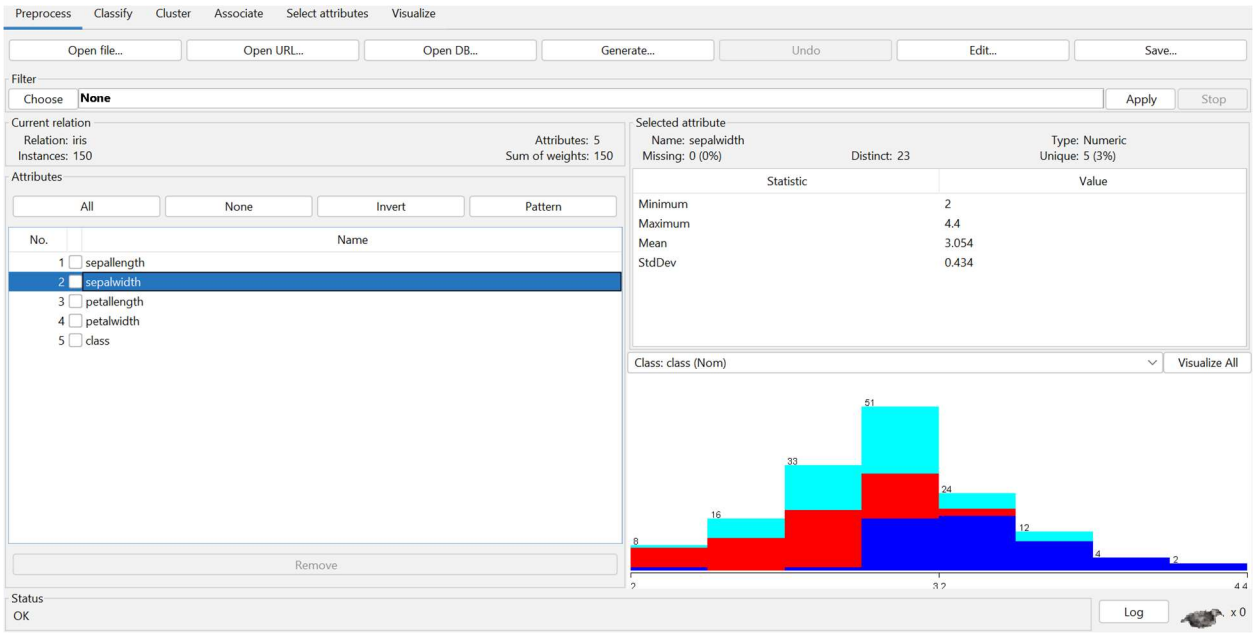


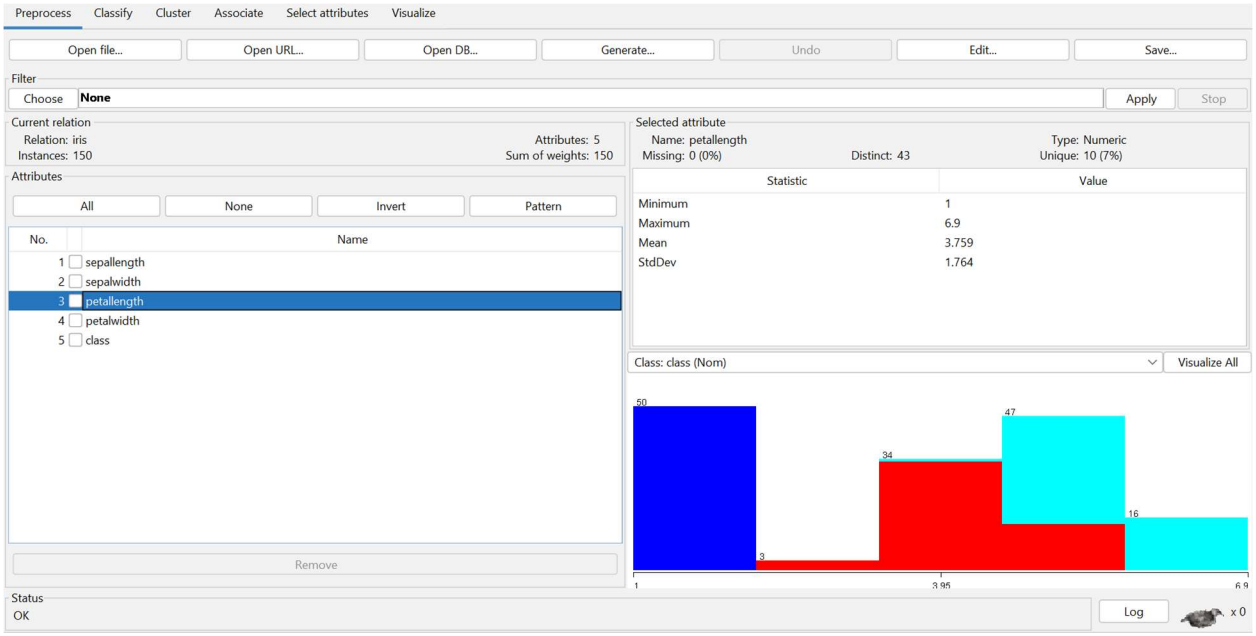
We utilized the Ranker attribute selection method within the Select Attribute tab in order to identify the attribute with the highest significance in the context of cluster formation or classification. Our analysis, employing the InfoGainAttributeEval method, has revealed that among all the attributes considered, Petal Length emerges as the most pivotal one. This finding underscores the critical role of Petal Length in shaping the clustering or classification outcomes.

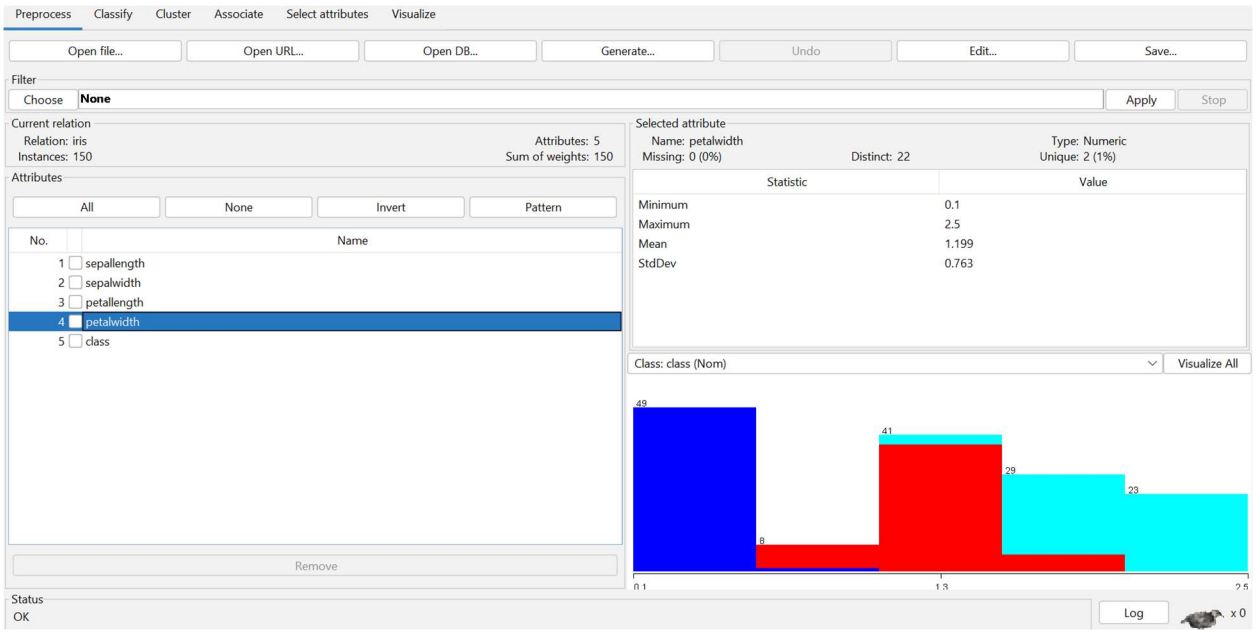
3. Preprocessing:

A.Y. 2023-2024

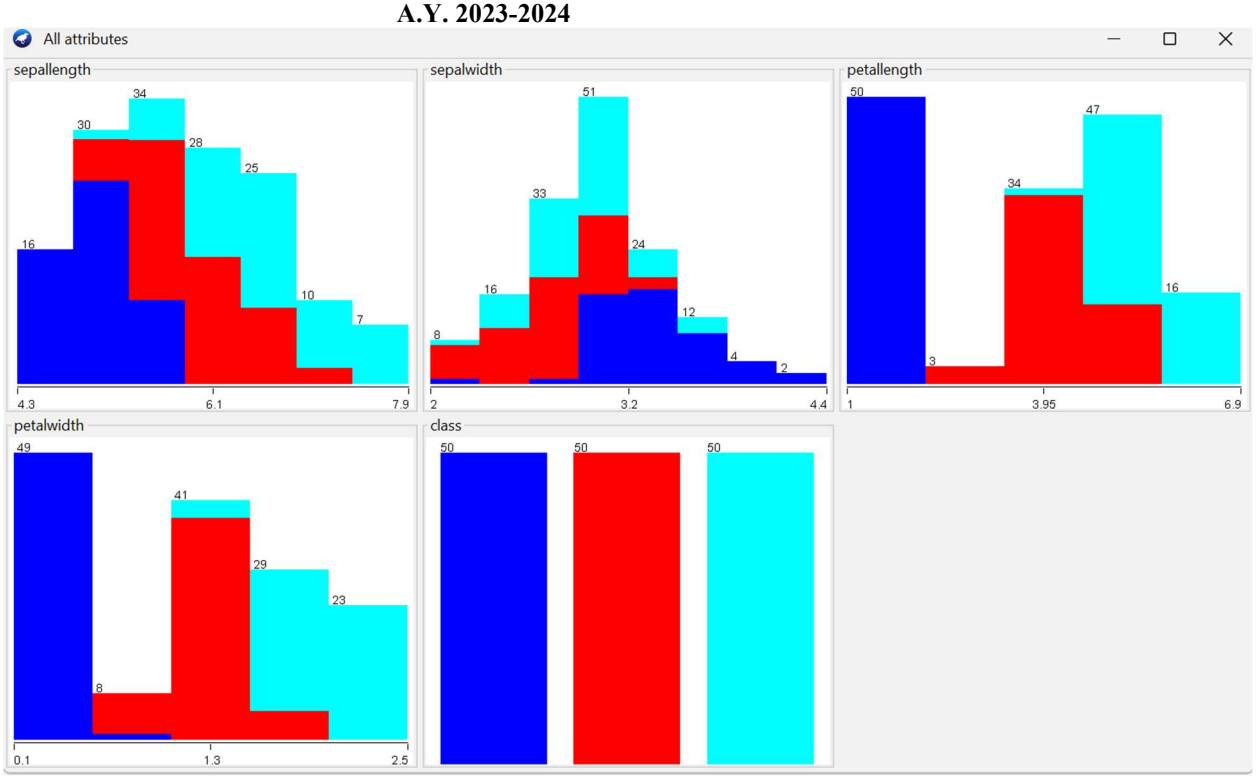

Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Open file...
Open URL...
Open DB...
Generate...
Undo
Edit...
Save...
Filter
Choose
None
Apply
Stop
Current relation
Relation: iris
Instances: 150
Attributes: 5
Sum of weights: 150
Attributes
All
None
Invert
Pattern
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Remove
Selected attribute
Name: sepallength
Missing: 0 (0%)
Distinct: 35
Type: Numeric
Unique: 9 (6%)
Statistic
Value
Minimum 4.3
Maximum 7.9
Mean 5.843
StdDev 0.828
Class: class (Nom)
Visualize All
34
30
28
25
16
10
7
4.3
6.1
7.9
Status
OK
Log
x 0


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: iris
Instances: 150
Attributes: 5
Sum of weights: 150
Attributes
All
None
Invert
Pattern
No.
Name
1 sepallength
2 sepalwidth
3 petallength
4 petalwidth
5 class
Remove
Selected attribute
Name: sepalwidth
Missing: 0 (0%)
Distinct: 23
Type: Numeric
Unique: 5 (3%)
Statistic
Value
Minimum 2
Maximum 4.4
Mean 3.054
StdDev 0.434
Class: class (Nom)
Visualize All
51
33
24
16
12
8
4
2
2
3.2
4.4
Status
OK
Log
x 0



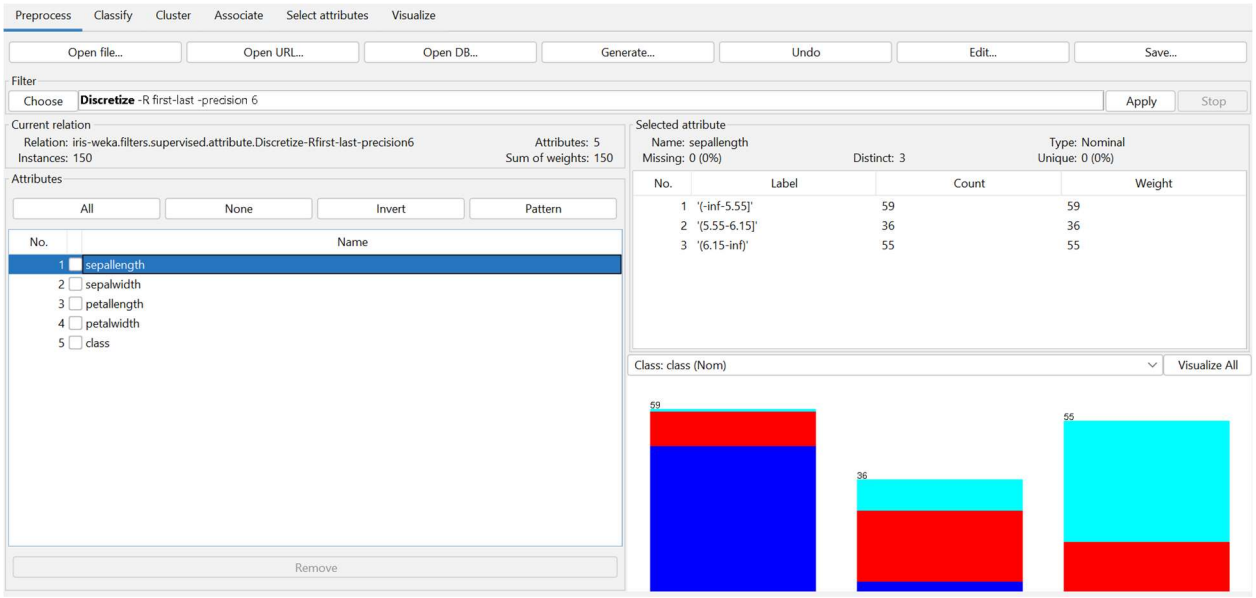


a. Visualize All: Select this button to visualize histograms of all attributes.



b. Filter: Choose Discretization under Unsupervised and Supervised methods. Observe the discretization and the outliers.

DISCRETIZATION:



c. IQR: Observe the IQR values for a selected attribute. Observe the outlier and

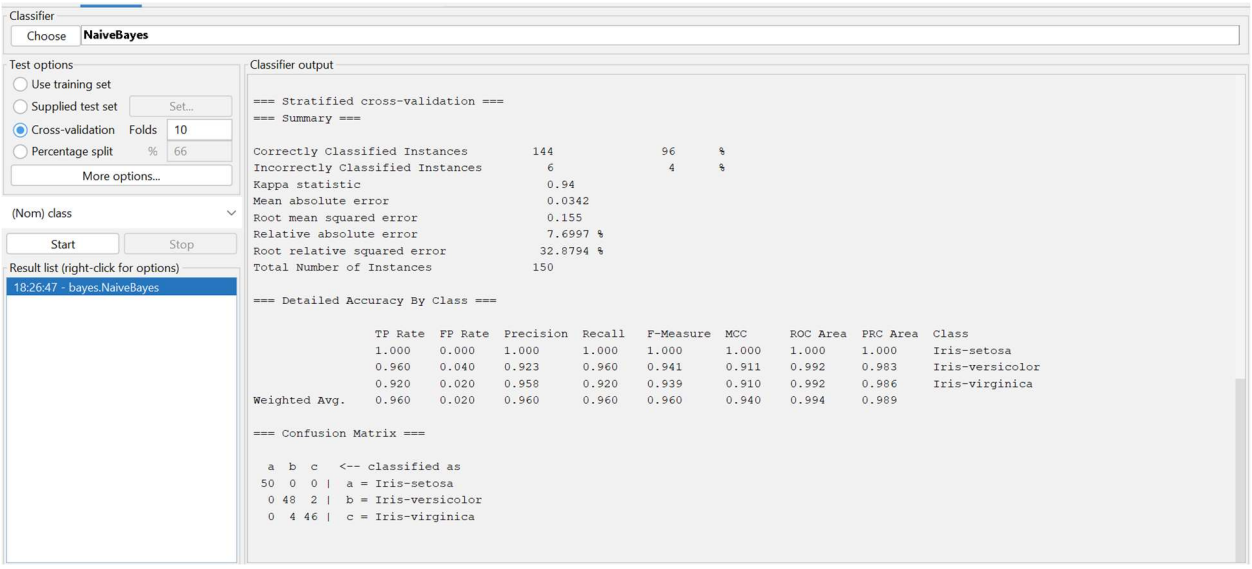
extreme values

d. **Removethevalue:** Remove instances with outlier values and show the screenshots of dataset before and after the removal.

4. **Classification:** Perform NB, kNN and DT/rule based classification

The "Classify" tab serves as a central hub for training and assessing the performance of various machine learning algorithms for both classification and regression tasks. These algorithms are grouped based on their respective characteristics and functionalities. The outcomes of these algorithm evaluations are stored in a result list, and a comprehensive summary of their performance is presented in the primary Classifier output.

In this specific instance, we are utilizing the Naive Bayes Classifier as one of the algorithms under assessment**.**



A.Y. 2023-2024

Classifier

Choose **NaiveBayes**

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 70

More options...

(Nom) class

Start | Stop

Result list (right-click for options)

18:26:47 - bayes.NaiveBayes
18:27:36 - bayes.NaiveBayes

Classifier output

```
Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          43               95.5556 %
Incorrectly Classified Instances         2                4.4444 %
Kappa statistic                          0.9331
Mean absolute error                      0.0375
Root mean squared error                  0.158
Relative absolute error                  8.422  %
Root relative squared error             33.4987 %
Total Number of Instances               45

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Iris-setosa
                 1.000    0.069    0.889      1.000    0.941      0.910    0.987     0.976     Iris-versicolor
                 0.867    0.000    1.000      0.867    0.929      0.901    0.987     0.979     Iris-virginica
Weighted Avg.    0.956    0.025    0.960      0.956    0.955      0.935    0.991     0.984

=== Confusion Matrix ===

  a  b  c   <-- classified as
 14  0  0 |  a = Iris-setosa
  0 16  0 |  b = Iris-versicolor
  0  2 13 |  c = Iris-virginica
```

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

Classifier

Choose **NaiveBayes**

Test options

- Use training set
- Supplied test set — Set...
- Cross-validation — Folds 10
- Percentage split — % 75

More options...

(Nom) class

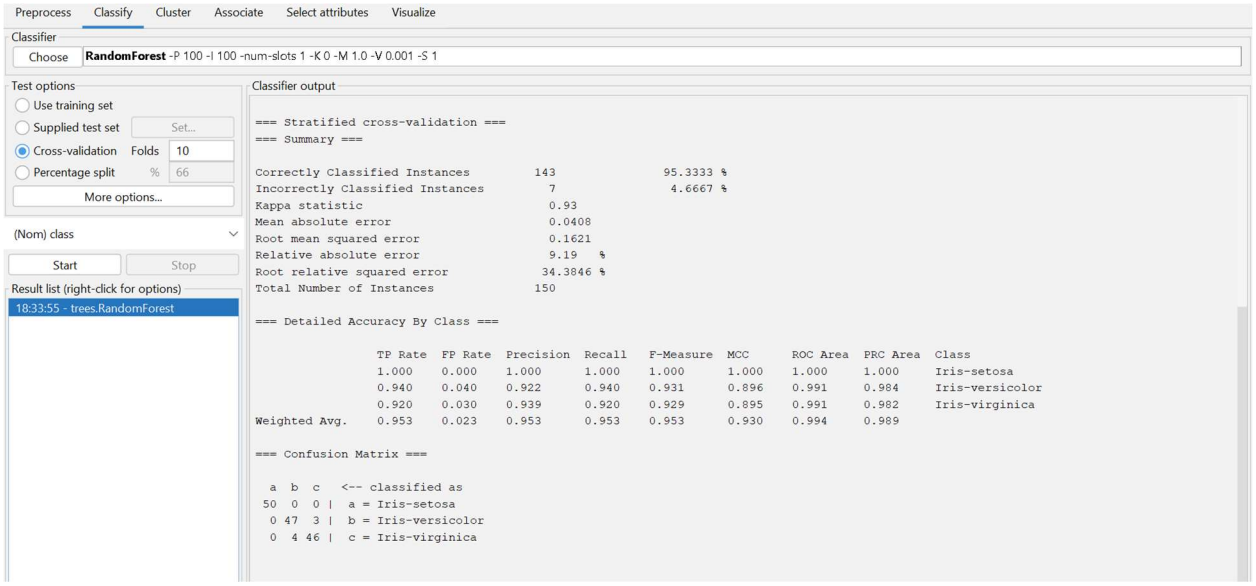
Start | Stop

Result list (right-click for options)

18:26:47 - bayes.NaiveBayes
18:27:36 - bayes.NaiveBayes
18:28:15 - bayes.NaiveBayes

Classifier output

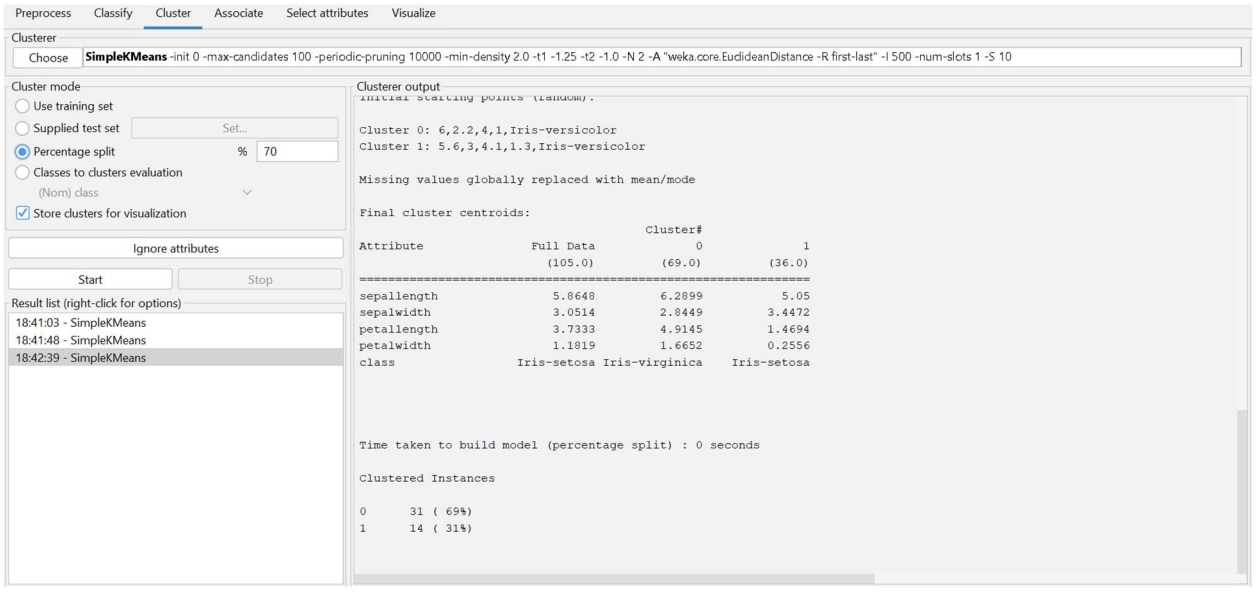
```
Time taken to test model on test split: 0 seconds

=== Summary ===

Correctly Classified Instances          35               94.5946 %
Incorrectly Classified Instances         2                5.4054 %
Kappa statistic                          0.9187
Mean absolute error                      0.0462
Root mean squared error                  0.1763
Relative absolute error                 10.3886 %
Root relative squared error             37.3672 %
Total Number of Instances               37

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Iris-setosa
                 1.000    0.083    0.867      1.000    0.929      0.891    0.981     0.963     Iris-versicolor
                 0.846    0.000    1.000      0.846    0.917      0.884    0.981     0.973     Iris-virginica
Weighted Avg.    0.946    0.029    0.953      0.946    0.946      0.921    0.986     0.977

=== Confusion Matrix ===

  a  b  c   <-- classified as
 11  0  0 |  a = Iris-setosa
  0 13  0 |  b = Iris-versicolor
  0  2 11 |  c = Iris-virginica
```

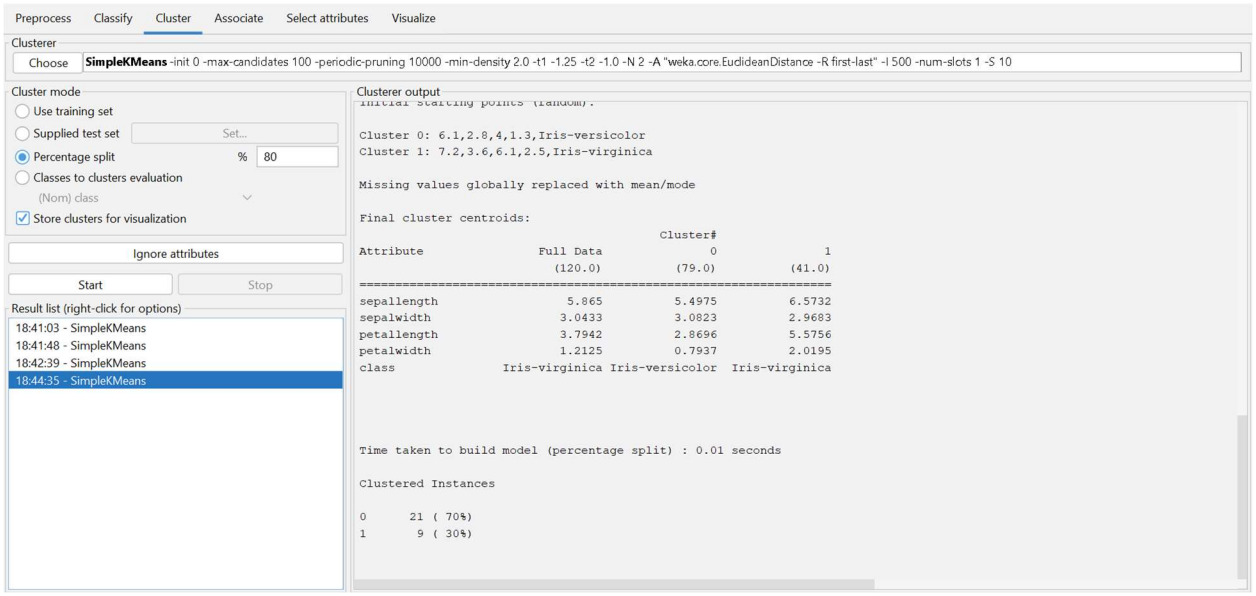


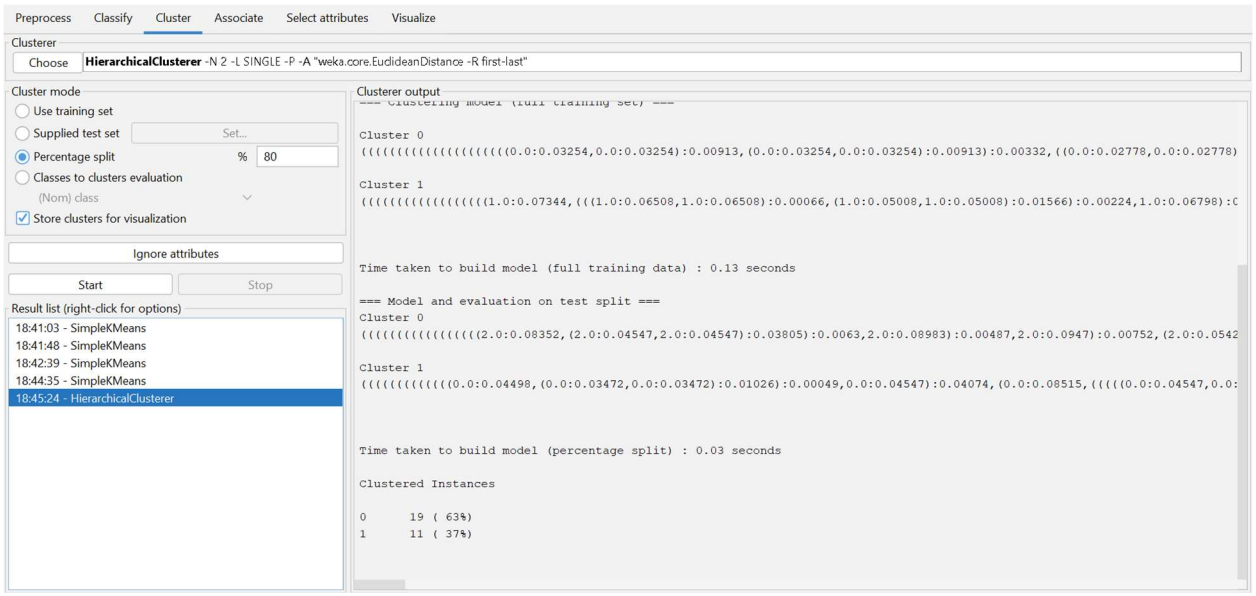
5. Clustering: Perform kmeans, hierarchical clustering and explain the output

The cluster tab is for training and evaluating the performance of different unsupervised clustering algorithms on your unlabeled dataset. Like the Classify tab, algorithms are divided into groups, results are kept in a result list and summarized in the main Clusterer output.

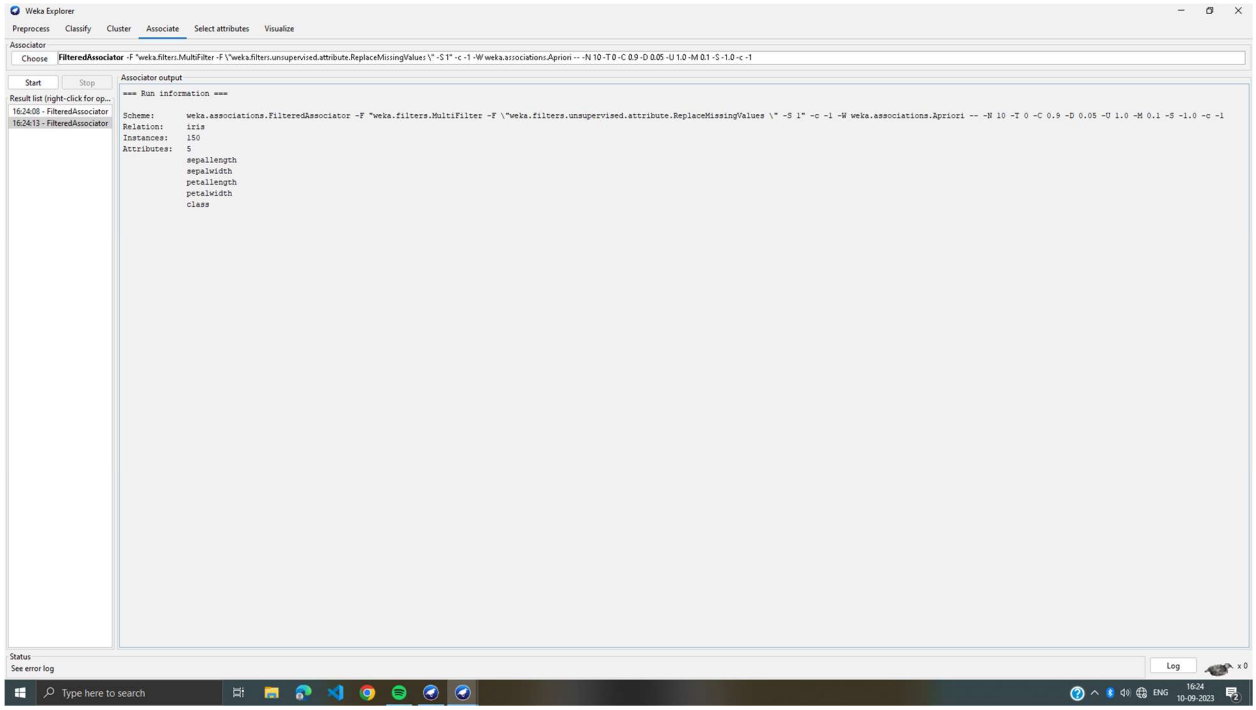
Here we are applying SimpleKmeans Clustering algorithm with 3 classes







6. **Association rule mining:** Perform apriori algo and show the rules created

7. 

Conclusion:

During our exploration of the Weka tool, we delved into the intricacies of data analysis. Our journey led us to work with two distinct databases: one focused on Iris petals, and the other on Supermarket data.

In this learning experience, we ventured into the realms of both supervised and unsupervised learning algorithms. What added richness to our analysis was the ability to visualize data transformations using various filtering techniques, offering us valuable insights into our datasets. To ascertain the most influential attribute for classification, we harnessed the power of the "select attribute" functionality, allowing us to rank attributes for their significance.

Furthermore, our exploration extended to implementing diverse clustering and classification algorithms, broadening our understanding of how these techniques can be applied to real-world datasets.

In the case of the Supermarket database, we took a fascinating dive into association rule mining by configuring the Apriori algorithm. This allowed us to uncover hidden patterns and relationships within the data, a process commonly referred to as market-basket analysis. Through these endeavors, we gained valuable hands-on experience in utilizing Weka as a versatile tool for data analysis and exploration.