

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:

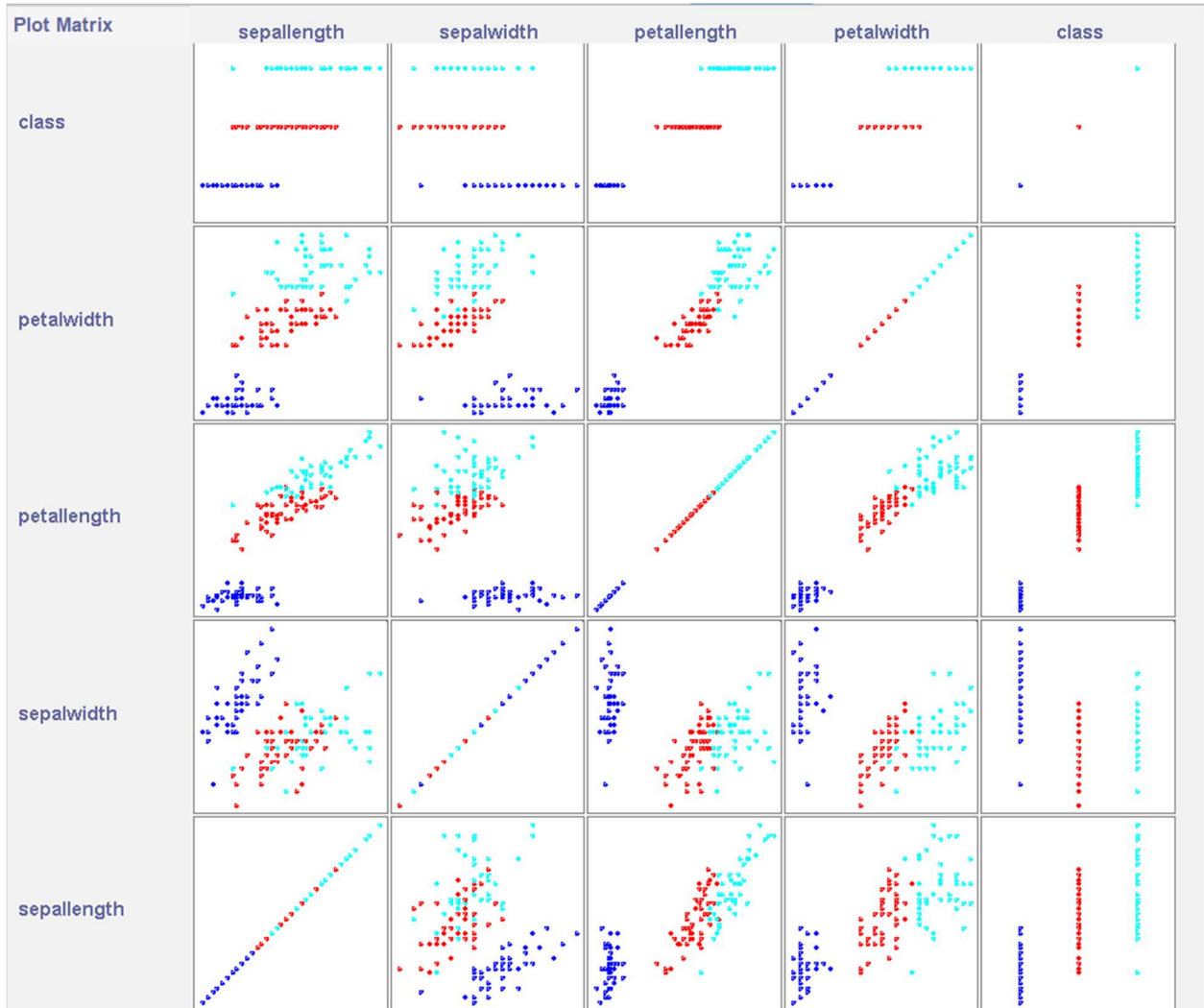
Visualization:

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



A.Y. 2023-2024



Upon observing the scatter plot in the visualize section, we can observe certain correlations within

the attributes. Some of them have been listed below

- Petal length vs Sepal length: Positive correlation
- Petal length vs Petal width: Positive correlation
- Petal width vs Sepal length: Positive correlation

- 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.



A.Y. 2023-2024

Preprocess Classify Cluster Associate **Select attributes** Visualize

Attribute Evaluator
Choose **InfoGainAttributeEval**

Search Method
Choose **Ranker** -T -1.7976931348623157E308 -N -1

Attribute Selection Mode
☒ Use full training set Folds: 10 Seed: 1
☐ Cross-validation

No class
Start Stop

Result list (right-click for options)
18:12:02 - Ranker + InfoGainAttributeEval

Attribute selection output
Instances: 150
Attributes: 5
sepal.length
sepal.width
petal.length
petal.width
class
Evaluation mode: evaluate on all training data
==== Attribute Selection on all input data ====
Search Method:
Attribute ranking.
Attribute Evaluator (supervised, Class (nominal): 5 class):
Information Gain Ranking Filter
Ranked attributes:
1.418 3 petal.length
1.378 4 petal.width
0.698 1 sepal.length
0.376 2 sepal.width
Selected attributes: 3,4,1,2 : 4

Status
OK

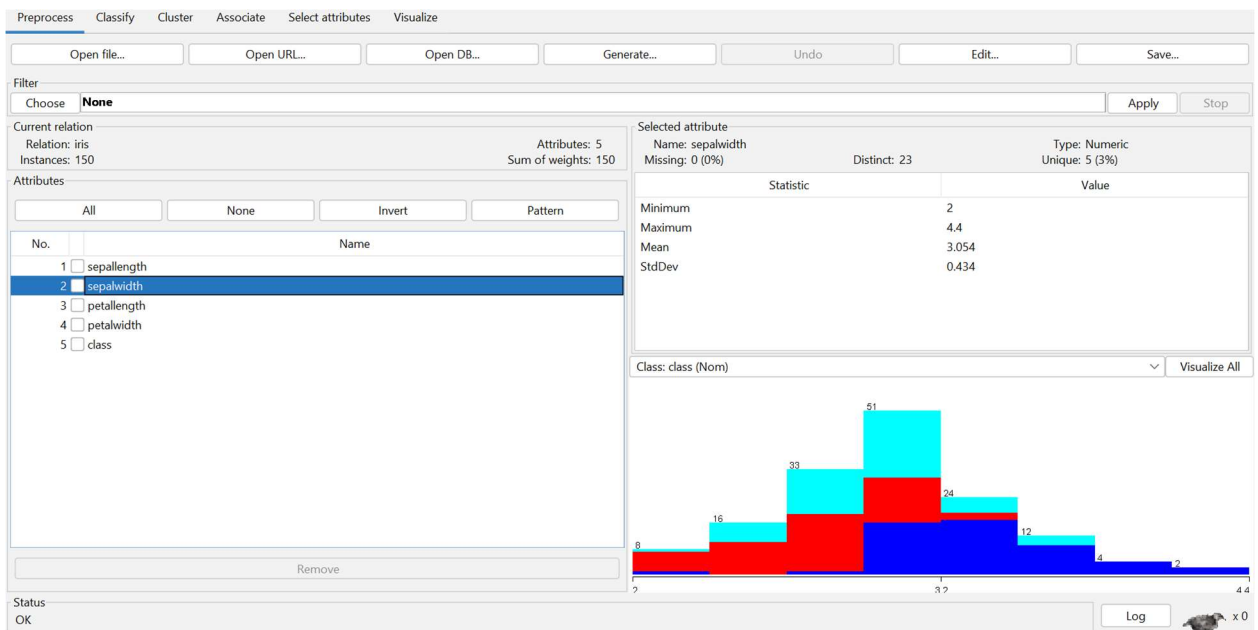
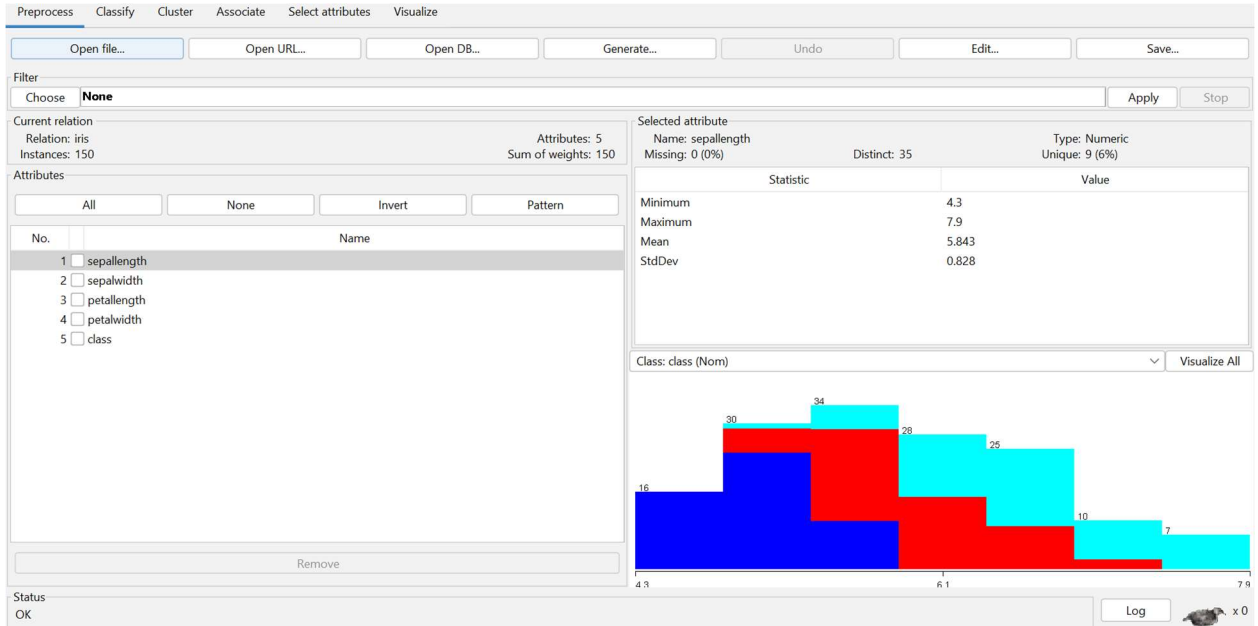
Log x 0

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:

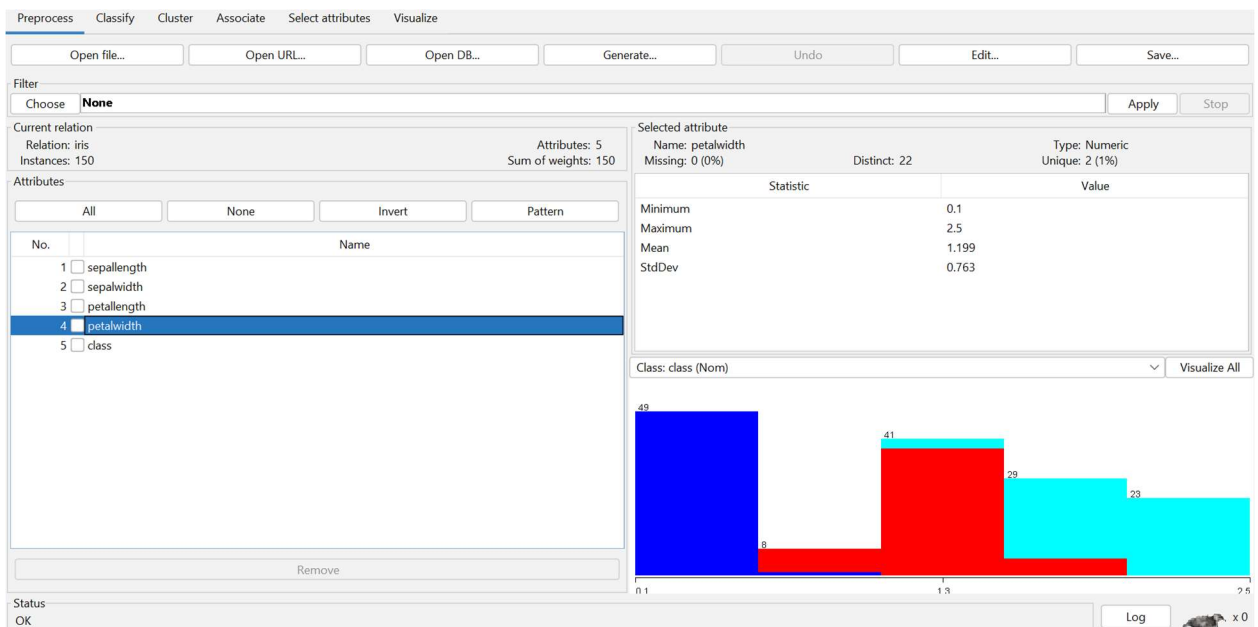
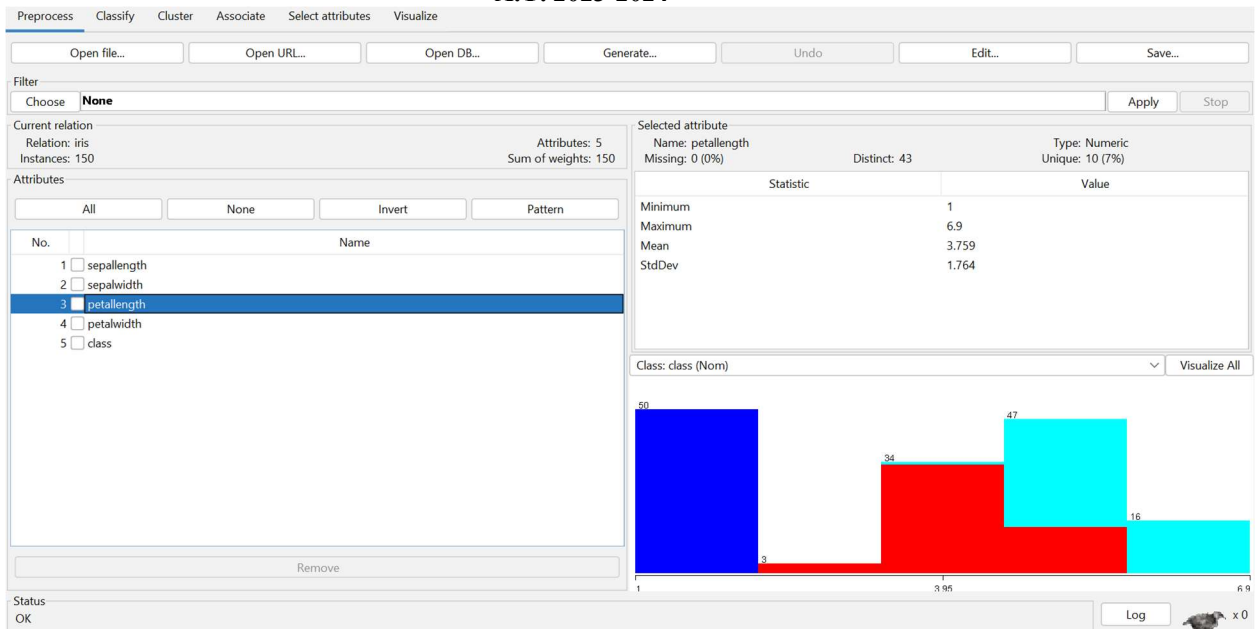


A.Y. 2023-2024





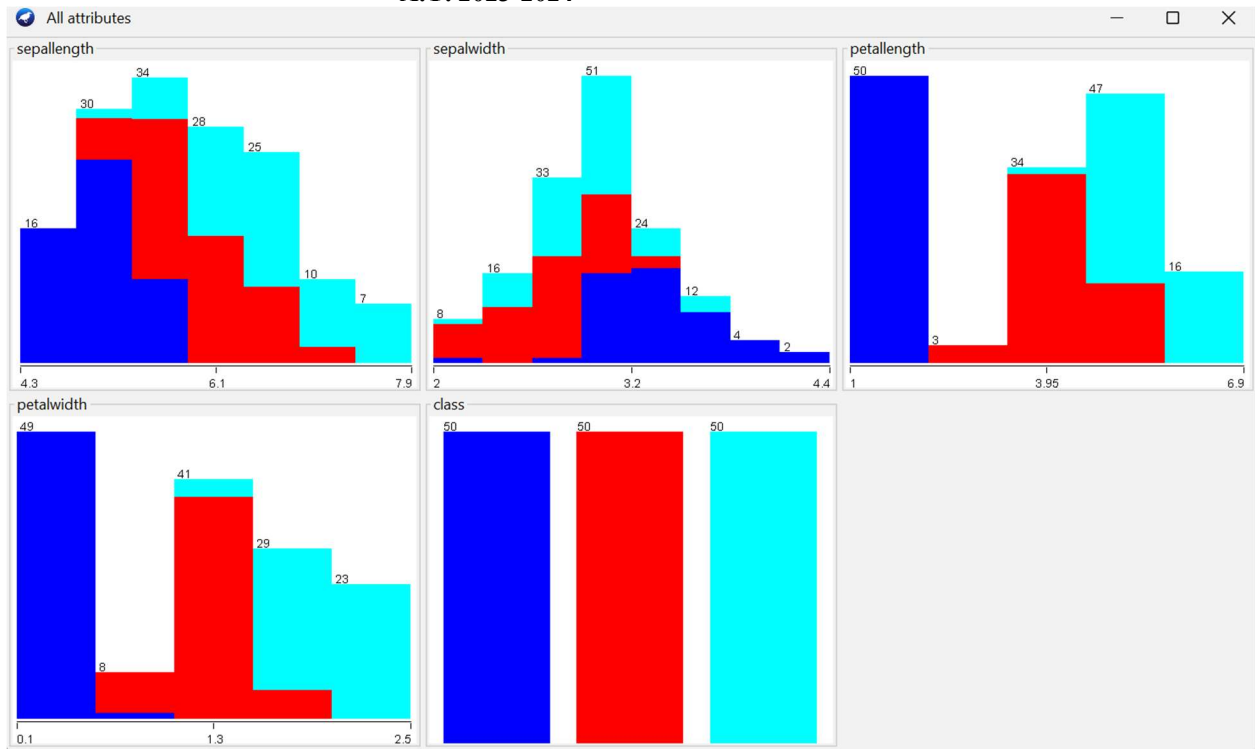
A.Y. 2023-2024



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



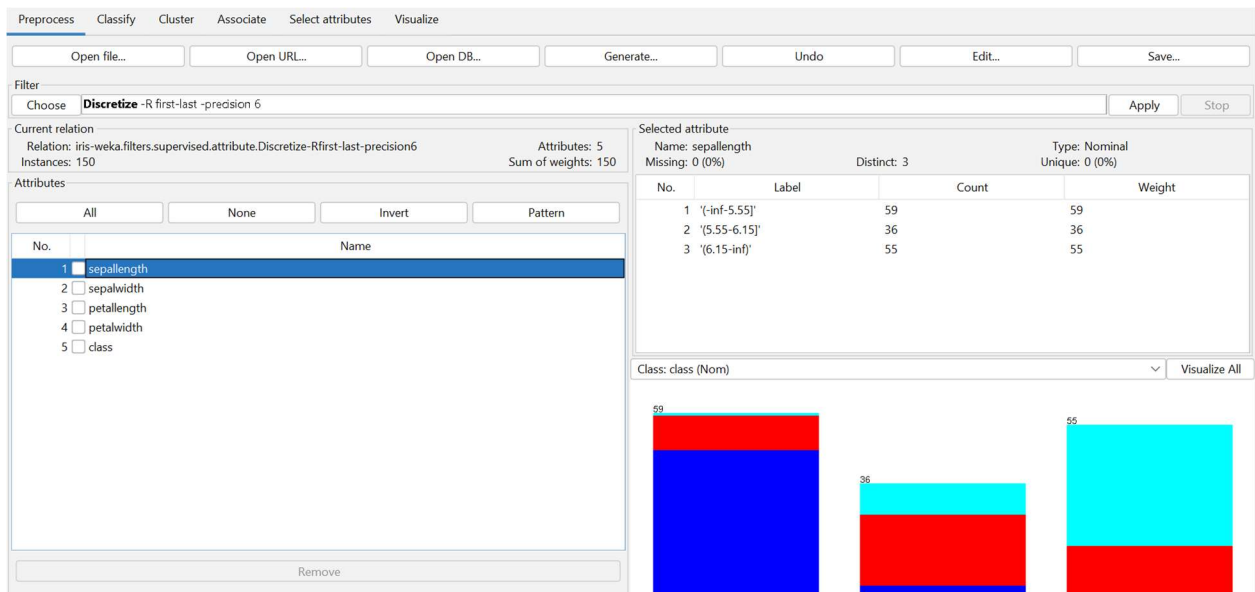
A.Y. 2023-2024



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



A.Y. 2023-2024

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.

The screenshot displays the Naive Bayes Classifier interface. On the left, the 'Test options' panel shows 'Cross-validation' selected with 10 folds and 66% split. The 'Classifier output' panel on the right provides a summary of performance metrics and a detailed accuracy table.

Classifier output Summary:

Metric	Value	Percentage
Correctly Classified Instances	144	96 %
Incorrectly Classified Instances	6	4 %
Kappa statistic	0.94	
Mean absolute error	0.0342	
Root mean squared error	0.155	
Relative absolute error	7.6997 %	
Root relative squared error	32.8794 %	
Total Number of Instances	150	

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
	0.960	0.040	0.923	0.960	0.941	0.911	0.992	0.983	Iris-versicolor
	0.920	0.020	0.958	0.920	0.939	0.910	0.992	0.986	Iris-virginica
Weighted Avg.	0.960	0.020	0.960	0.960	0.960	0.940	0.994	0.989	

Confusion Matrix:

```
a b c <-- classified as
50 0 0 | a = Iris-setosa
0 48 2 | b = Iris-versicolor
0 4 46 | c = Iris-virginica
```



A.Y. 2023-2024

Classifier

Choose **NaiveBayes**

Test options

☐ Use training set

☐ Supplied test set

☐ Cross-validation Folds 10

☒ Percentage split % 70

(Nom) class

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

☐ Cross-validation Folds 10

☒ Percentage split % 75

(Nom) class

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
```



A.Y. 2023-2024

Preprocess **Classify** Cluster Associate Select attributes Visualize

Classifier
Choose **RandomForest** -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1

Test options
☐ Use training set
☐ Supplied test set Set...
☒ Cross-validation Folds **10**
☐ Percentage split % 66
More options...

(Nom) class
Start Stop

Result list (right-click for options)
18:33:55 - trees.RandomForest

Classifier output

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

Correctly Classified Instances      143          95.3333 %
Incorrectly Classified Instances     7           4.6667 %
Kappa statistic                    0.93
Mean absolute error                 0.0408
Root mean squared error             0.1621
Relative absolute error             9.19 %
Root relative squared error        34.3846 %
Total Number of Instances          150

=== 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
0.940  0.040  0.922  0.940  0.931  0.896  0.991  0.984  Iris-versicolor
0.920  0.030  0.939  0.920  0.929  0.895  0.991  0.982  Iris-virginica
Weighted Avg.  0.953  0.023  0.953  0.953  0.953  0.930  0.994  0.989

=== Confusion Matrix ===

  a  b  c  <-- classified as
50  0  0 | a = Iris-setosa
 0 47  3 | b = Iris-versicolor
 0  4 46 | 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

Preprocess **Classify** **Cluster** Associate Select attributes Visualize

Clusterer
Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode
☐ Use training set
☐ Supplied test set Set...
☒ Percentage split % 70
☐ Classes to clusters evaluation
(Nom) class
☒ Store clusters for visualization

Ignore attributes
Start Stop

Result list (right-click for options)
18:41:03 - SimpleKMeans
18:41:48 - SimpleKMeans
18:42:39 - SimpleKMeans

Clusterer output

```
INITIAL SEEDING POINTS (RANDOM).

Cluster 0: 6,2,2,4,1,Iris-versicolor
Cluster 1: 5,6,3,4,1,1,3,Iris-versicolor

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute      Full Data      Cluster#
              (105.0)      (69.0)      (36.0)
=====
sepalength     5.8648         6.2899         5.05
sepalwidth     3.0514         2.8449         3.4472
petallength    3.7333         4.9145         1.4694
petalwidth     1.1819         1.6652         0.2556
class          Iris-setosa Iris-virginica Iris-setosa

Time taken to build model (percentage split) : 0 seconds

Clustered Instances

0      31 ( 69%)
1      14 ( 31%)
```



A.Y. 2023-2024

Preprocess Classify **Cluster** Associate Select attributes Visualize

Clusterer
Choose **SimpleKMeans** -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 -S 10

Cluster mode
☐ Use training set
☐ Supplied test set Set...
☒ Percentage split % 80
☐ Classes to clusters evaluation (Nom) class
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 18:41:03 - SimpleKMeans
- 18:41:48 - SimpleKMeans
- 18:42:39 - SimpleKMeans
- 18:44:35 - SimpleKMeans

Clusterer output

Initial starting points (random):

Cluster 0: 6.1,2.8,4.1,3,Iris-versicolor
Cluster 1: 7.2,3.6,6.1,2.5,Iris-virginica

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute	Full Data (120.0)	Cluster# 0 (79.0)	1 (41.0)
sepal.length	5.865	5.4975	6.5732
sepal.width	3.0433	3.0823	2.9683
petal.length	3.7942	2.8696	5.5756
petal.width	1.2125	0.7937	2.0195
class		Iris-virginica	Iris-versicolor

Time taken to build model (percentage split) : 0.01 seconds

Clustered Instances

0	21 (70%)
1	9 (30%)

Preprocess Classify **Cluster** Associate Select attributes Visualize

Clusterer
Choose **HierarchicalClusterer** -N 2 -L SINGLE -P -A "weka.core.EuclideanDistance -R first-last"

Cluster mode
☐ Use training set
☐ Supplied test set Set...
☒ Percentage split % 80
☐ Classes to clusters evaluation (Nom) class
☒ Store clusters for visualization

Ignore attributes

Start Stop

Result list (right-click for options)

- 18:41:03 - SimpleKMeans
- 18:41:48 - SimpleKMeans
- 18:42:39 - SimpleKMeans
- 18:44:35 - SimpleKMeans
- 18:45:24 - HierarchicalClusterer

Clusterer output

==== Clustering modes (full training set) ===

Cluster 0
((((((((((((((((((0.0:0.03254,0.0:0.03254):0.00913,(0.0:0.03254,0.0:0.03254):0.00913):0.00332,((0.0:0.02778,0.0:0.02778)

Cluster 1
((((((((((((((((((1.0:0.07344,((1.0:0.06508,1.0:0.06508):0.00066,(1.0:0.05008,1.0:0.05008):0.01566):0.00224,1.0:0.06798):C

Time taken to build model (full training data) : 0.13 seconds

=== Model and evaluation on test split ===

Cluster 0
((((((((((((((((((2.0:0.08352,(2.0:0.04547,2.0:0.04547):0.03805):0.0063,2.0:0.08983):0.00487,2.0:0.0947):0.00752,(2.0:0.0542

Cluster 1
((((((((((((((((((0.0:0.04498,(0.0:0.03472,0.0:0.03472):0.01026):0.00049,0.0:0.04547):0.04074,(0.0:0.08515,(((0.0:0.04547,0.0:

Time taken to build model (percentage split) : 0.03 seconds

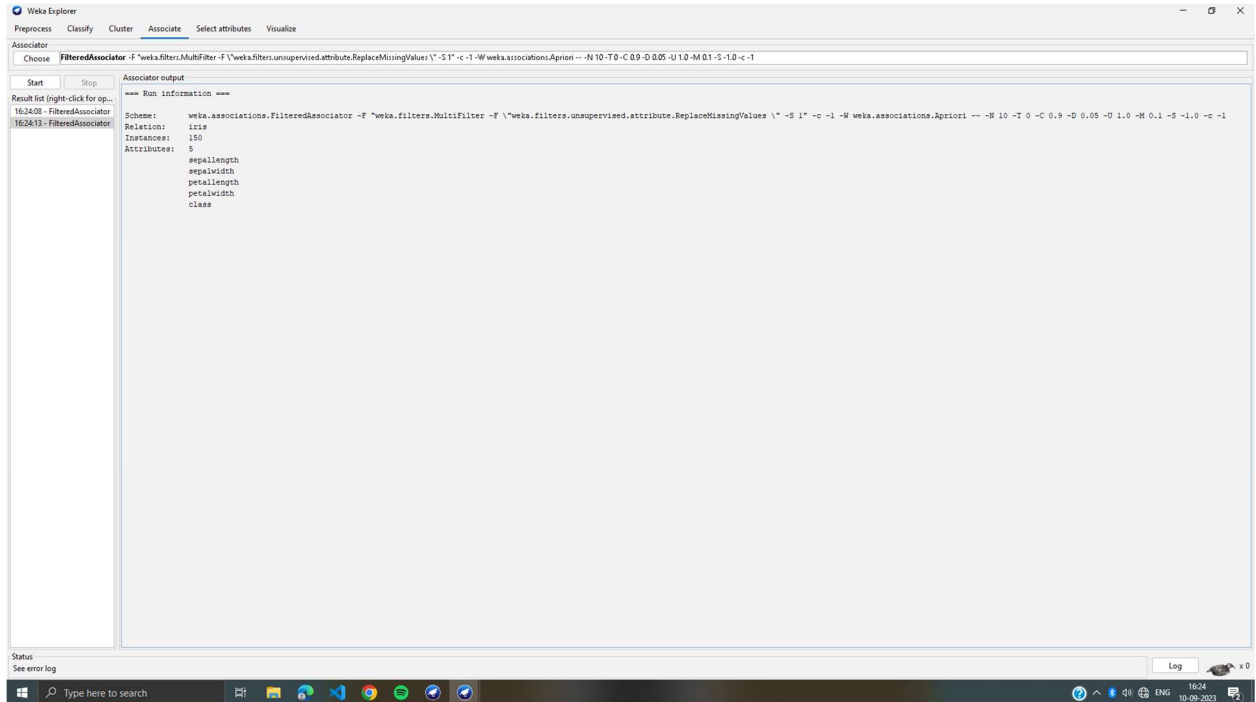
Clustered Instances

0	19 (63%)
1	11 (37%)

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



A.Y. 2023-2024



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.