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)

JUNAID GIRKAR

60004190057

TE Comps A4

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:

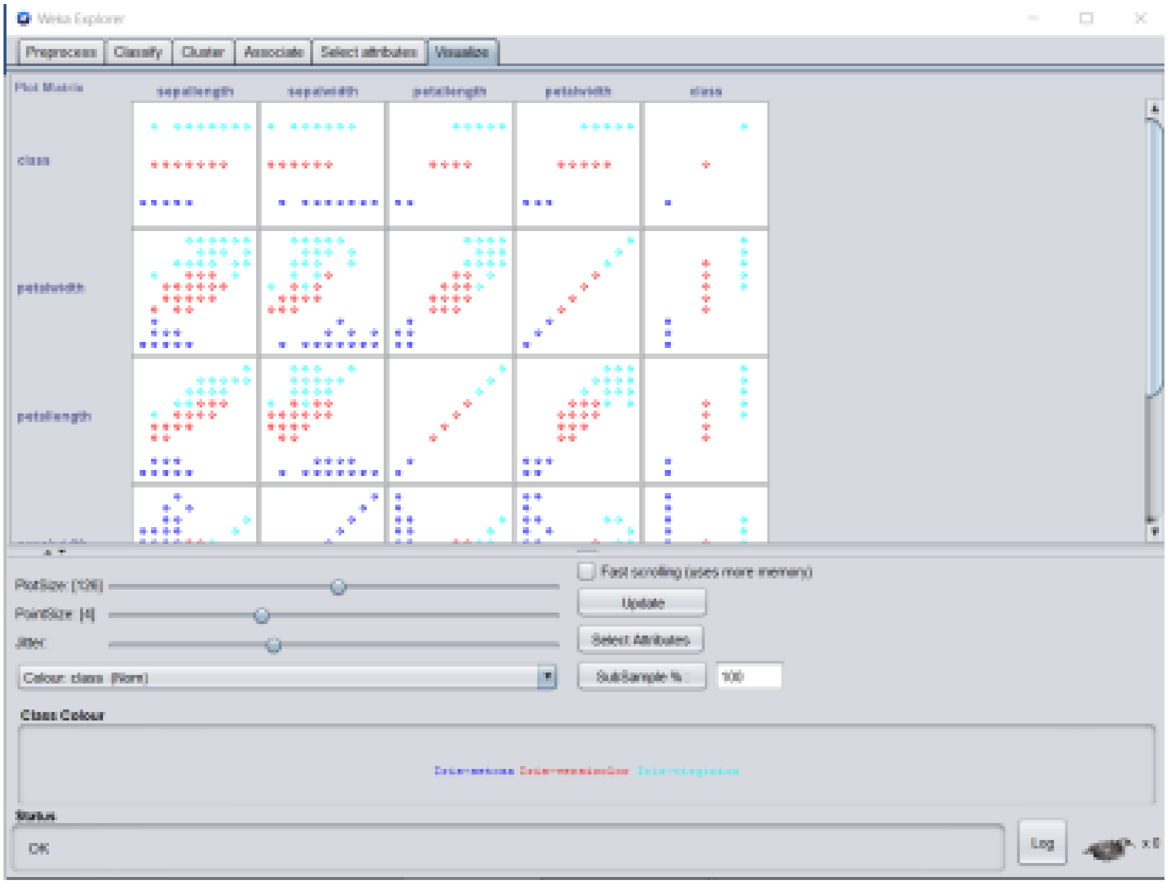
Visualization:

WEKA

The workbench for machine learning

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



☐ Fast scrolling (uses more memory)

PlotSize: [100]

PointSize: [4]

Jitter:

Colour: class (Nom)

SubSample % 100

OK

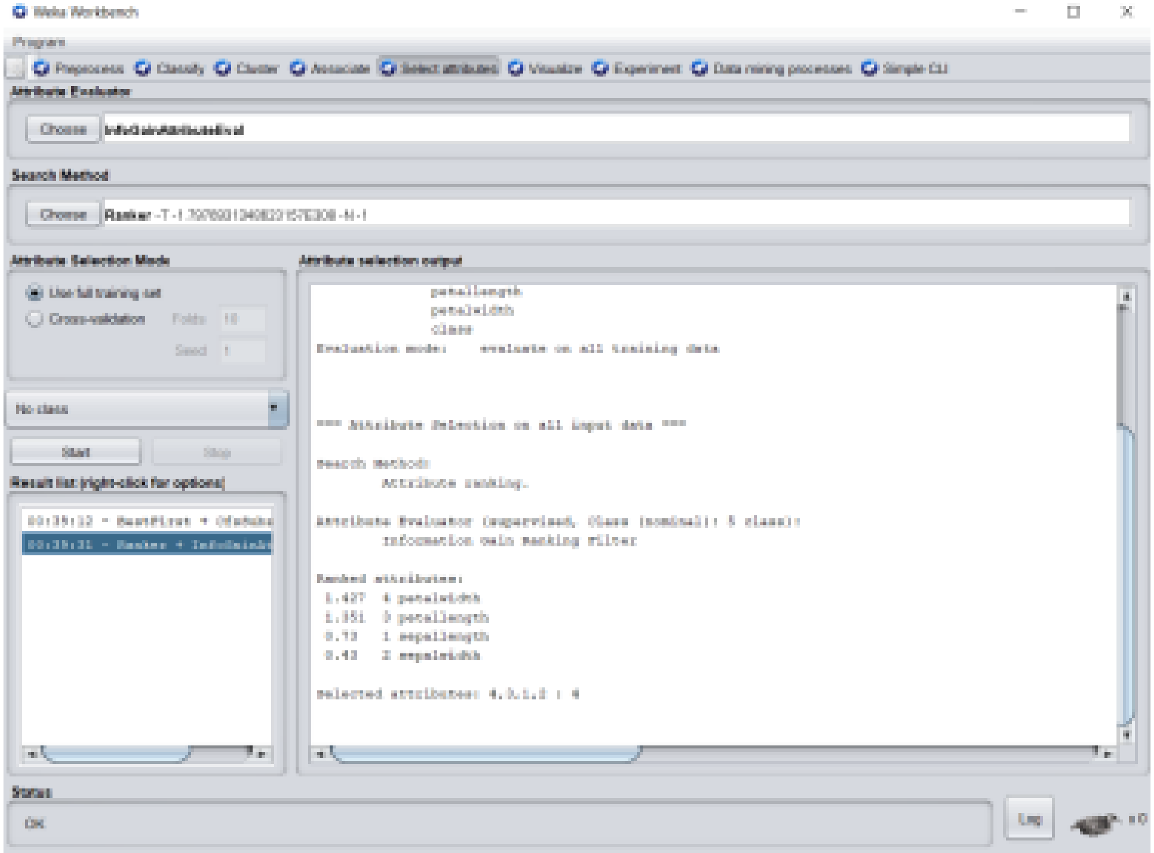
Upon observing the scatter plot in the visualize section, we can observe certain correlations within the attributes. Some of them have been listed below

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.



Attribute Evaluator: InfoGainAttributeEval

Search Method: Ranker -T -1.7976931348623157E308 -N -1

Attribute Selection Mode: ● Use full training set ○ Cross-validation Folds 10 Seed 1

No class

Result list (right-click for options):
00:35:12 - BestFirst + CfsSubs
00:35:31 - Ranker + InfoGainA

Attribute selection output:

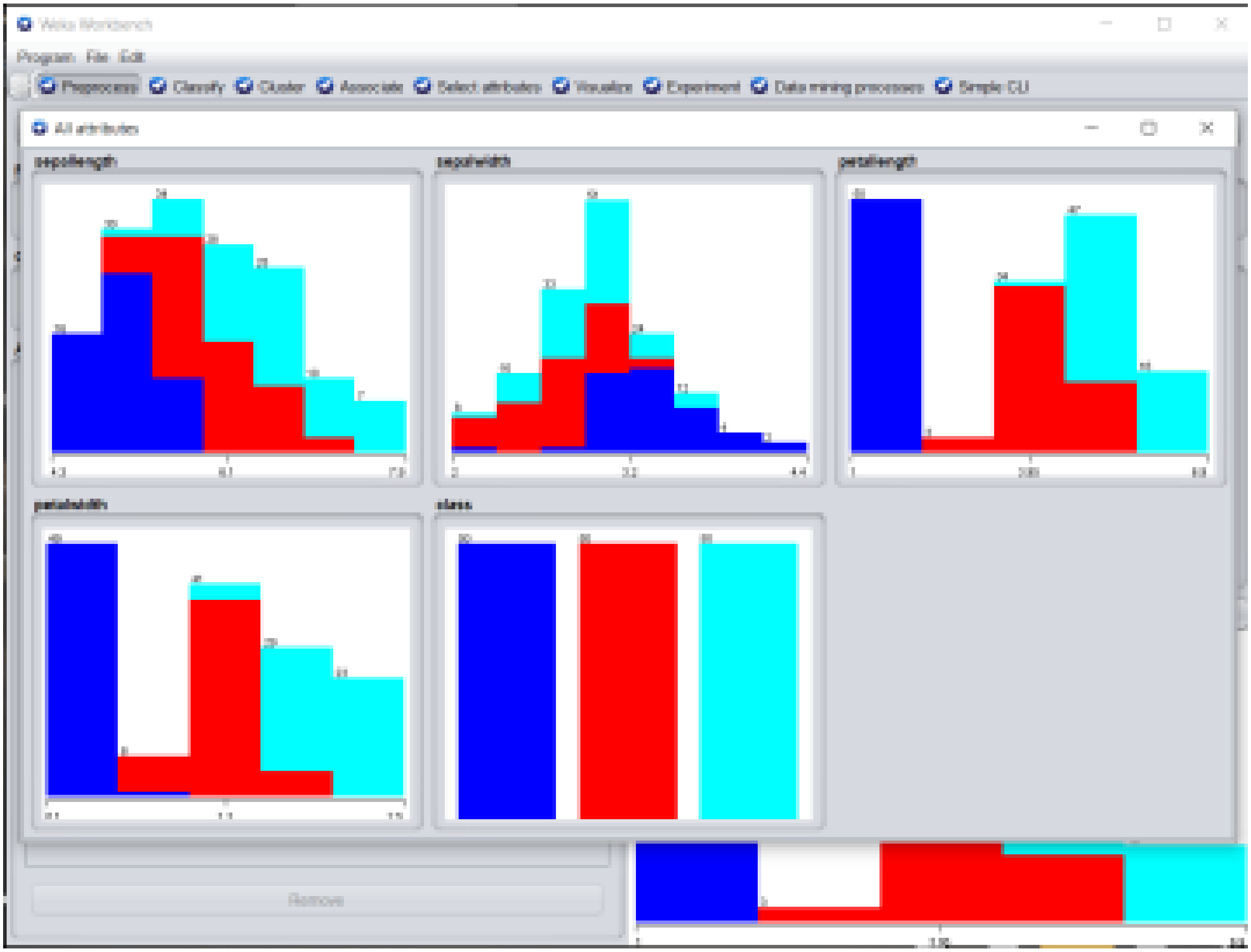
```
                    petallength
                    petalwidth
                    class
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 5 class):
        Information Gain Ranking Filter

Ranked attributes:
 1.427  4 petalwidth
 1.351  3 petallength
 0.72   1 sepallength
 0.43   2 sepalwidth

Selected attributes: 4,3,1,2 : 4
```

Status: OK

We apply the attribute Ranker using the InfoGainAttributeEval method in the Select Attribute tab to determine which attribute holds the most importance while forming clusters/classification. The results show that Petal Length is the most important attribute among the others

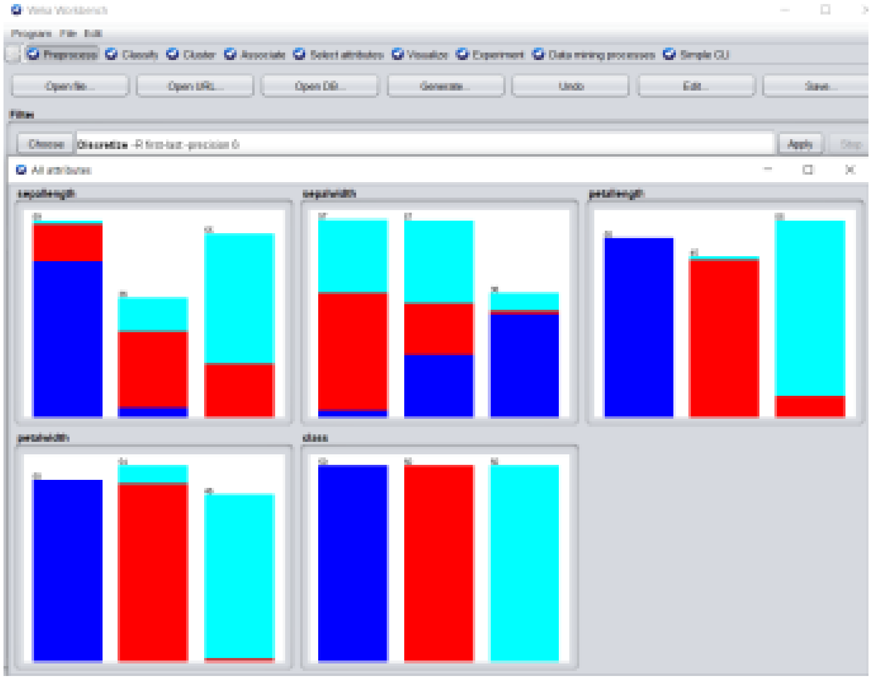
3. Preprocessing:

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.

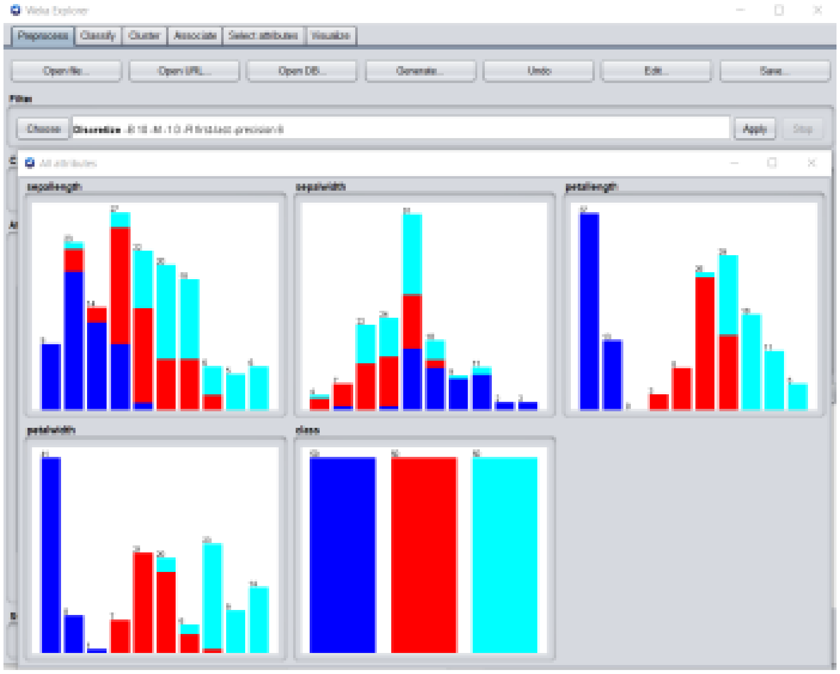
SUPERVISED DISCRETIZE FILTERING



Filter

Discretize -R first-last -precision 6

UNSUPERVISED DISCRETIZE FILTERING



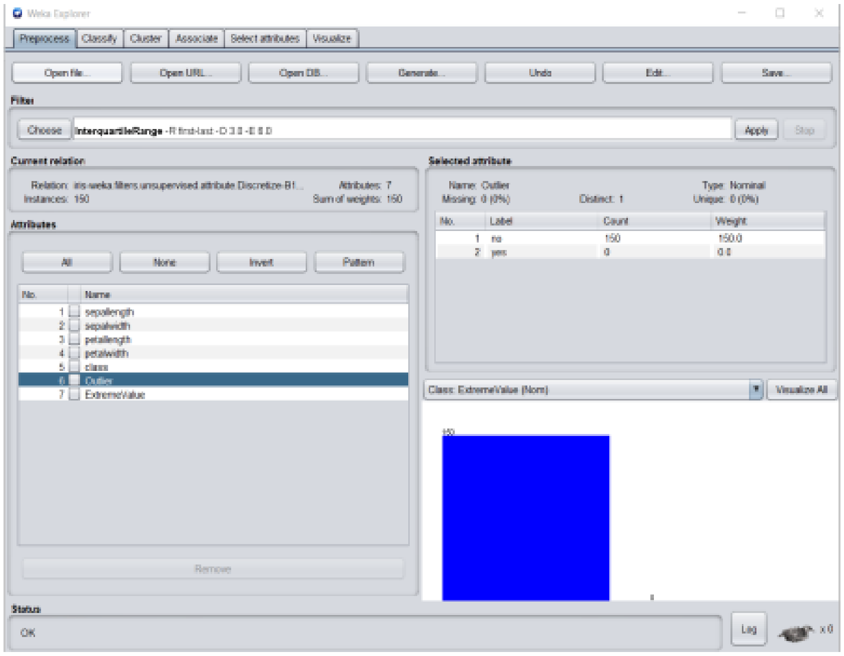
Filter

Discretize -B 10 -M -1.0 -R first-last -precision 6

We can observe the discretization of the Sepal Length variable. The number of bins reduces from 6 to 3.

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

OUTLIER



Filter: **InterquartileRange** -R first-last -O 3.0 -E 6.0

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Discretize-B1... Attributes: 7
Instances: 150 Sum of weights: 150

Selected attribute

Name: Outlier Type: Nominal
Missing: 0 (0%) Distinct: 1 Unique: 0 (0%)

No.	Label	Count	Weight
1	no	150	150.0
2	yes	0	0.0

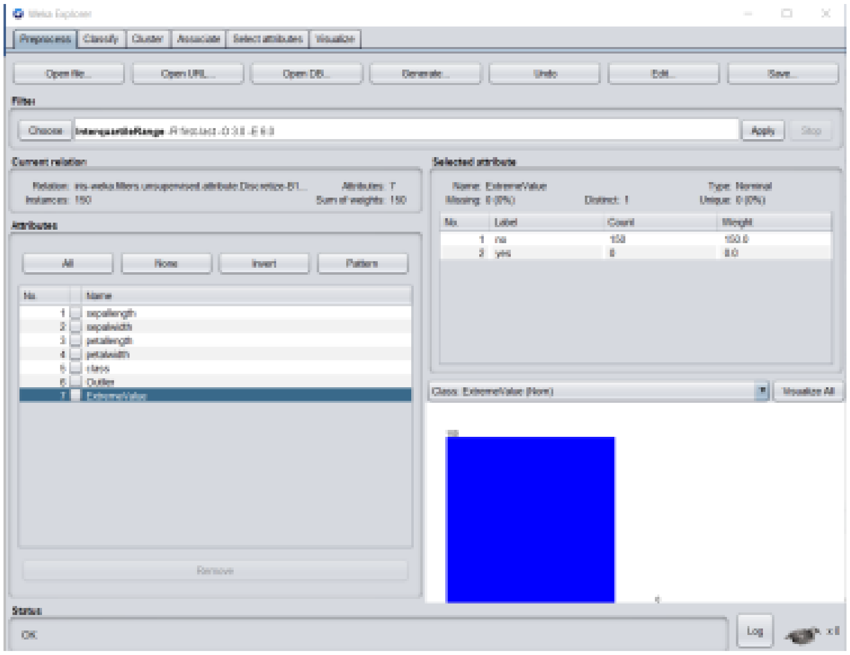
Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class
6	Outlier
7	ExtremeValue

Class: ExtremeValue (Nom)

Status: OK

EXTREME VALUE



Filter: **InterquartileRange** -R first-last -O 3.0 -E 6.0

Current relation

Relation: iris-weka.filters.unsupervised.attribute.Discretize-B1... Attributes: 7
Instances: 150 Sum of weights: 150

Selected attribute

Name: ExtremeValue Type: Nominal
Missing: 0 (0%) Distinct: 1 Unique: 0 (0%)

No.	Label	Count	Weight
1	no	150	150.0
2	yes	0	0.0

Attributes

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class
6	Outlier
7	ExtremeValue

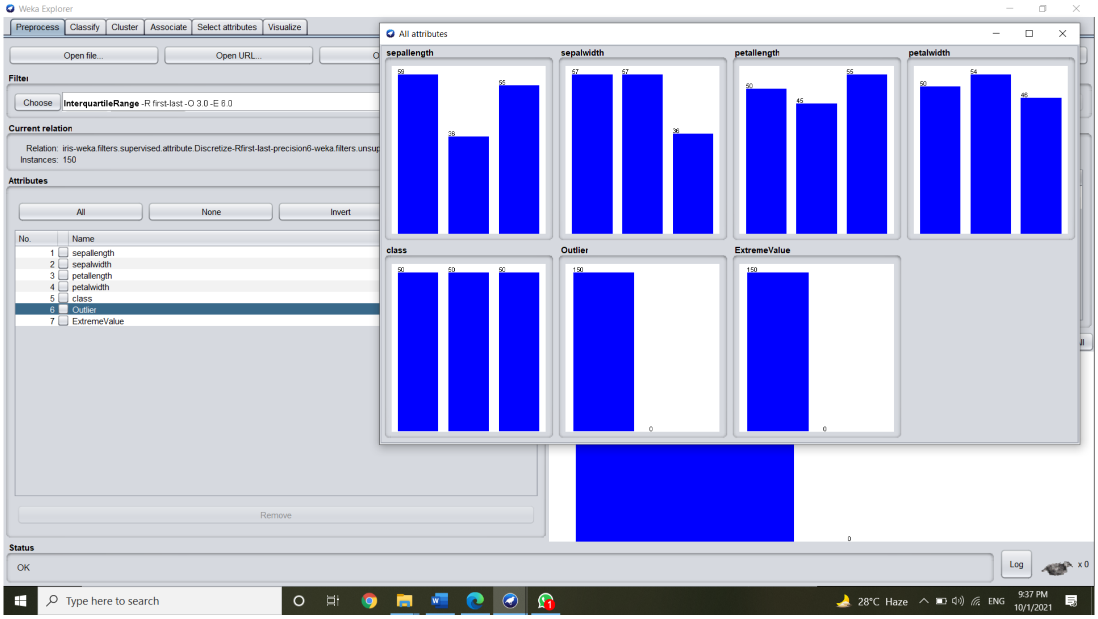
Class: ExtremeValue (Nom)

Status: OK

Upon visualizing all the variables, we can view the outliers present per variable and can remove the extreme values.

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

BEFORE DELETION:



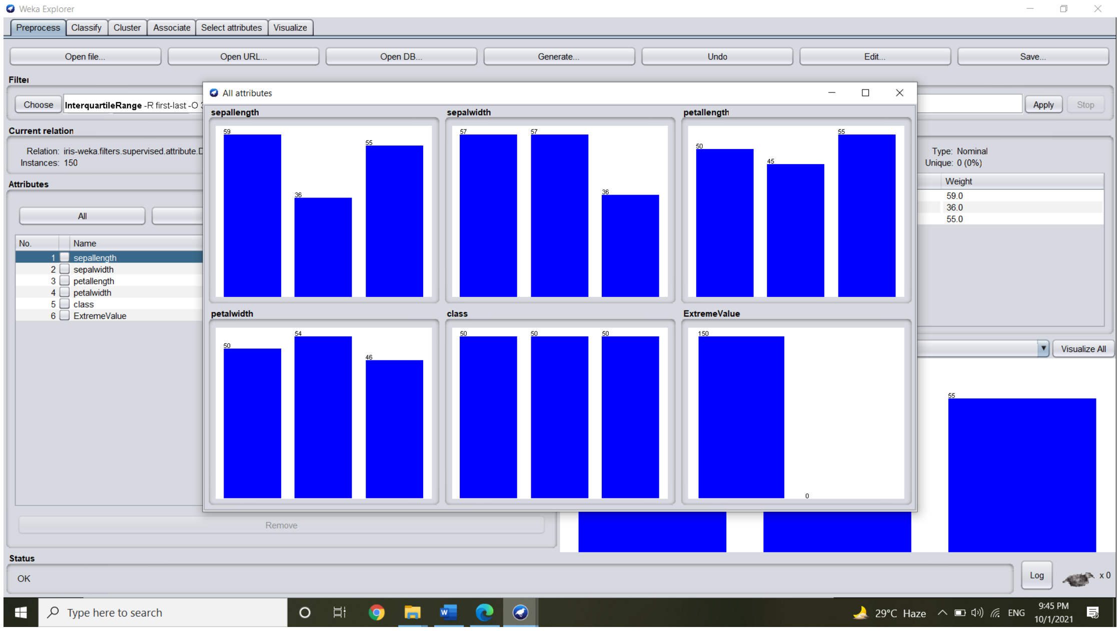
Filter: InterquartileRange -R first-last -O 3.0 -E 6.0

Relation: iris-weka.filters.supervised.attribute.Discretize-Rfirst-last-precision6-weka.filters.unsup
Instances: 150

No.	Name
1	sepallength
2	sepalwidth
3	petallength
4	petalwidth
5	class
6	Outlier
7	ExtremeValue

Status: OK

AFTER DELETION:



Filter: InterquartileRange -R first-last -O 3

Relation: iris-weka.filters.supervised.attribute.D
Instances: 150

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

Type: Nominal
Unique: 0 (0%)

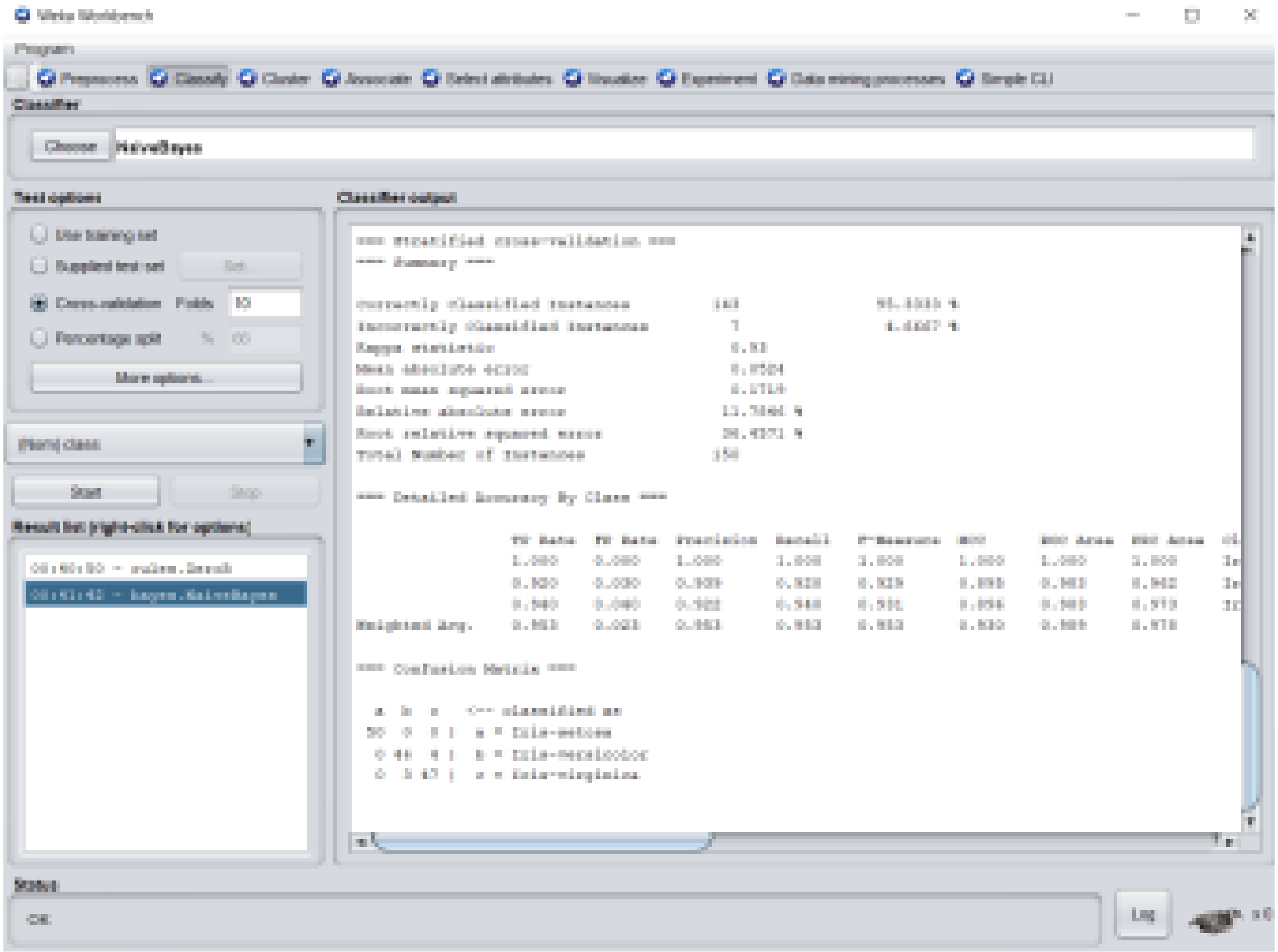
Weight
59.0
36.0
55.0

Status: OK

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

The **classify** tab is for training and evaluating the performance of different machine learning algorithms on your classification or regression problem. Algorithms are divided up into groups, results are kept in a result list and summarized in the main Classifier output.

Here we are applying the Naive Bayers Classifier.



Classifier: NaiveBayes

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

(Nom) class

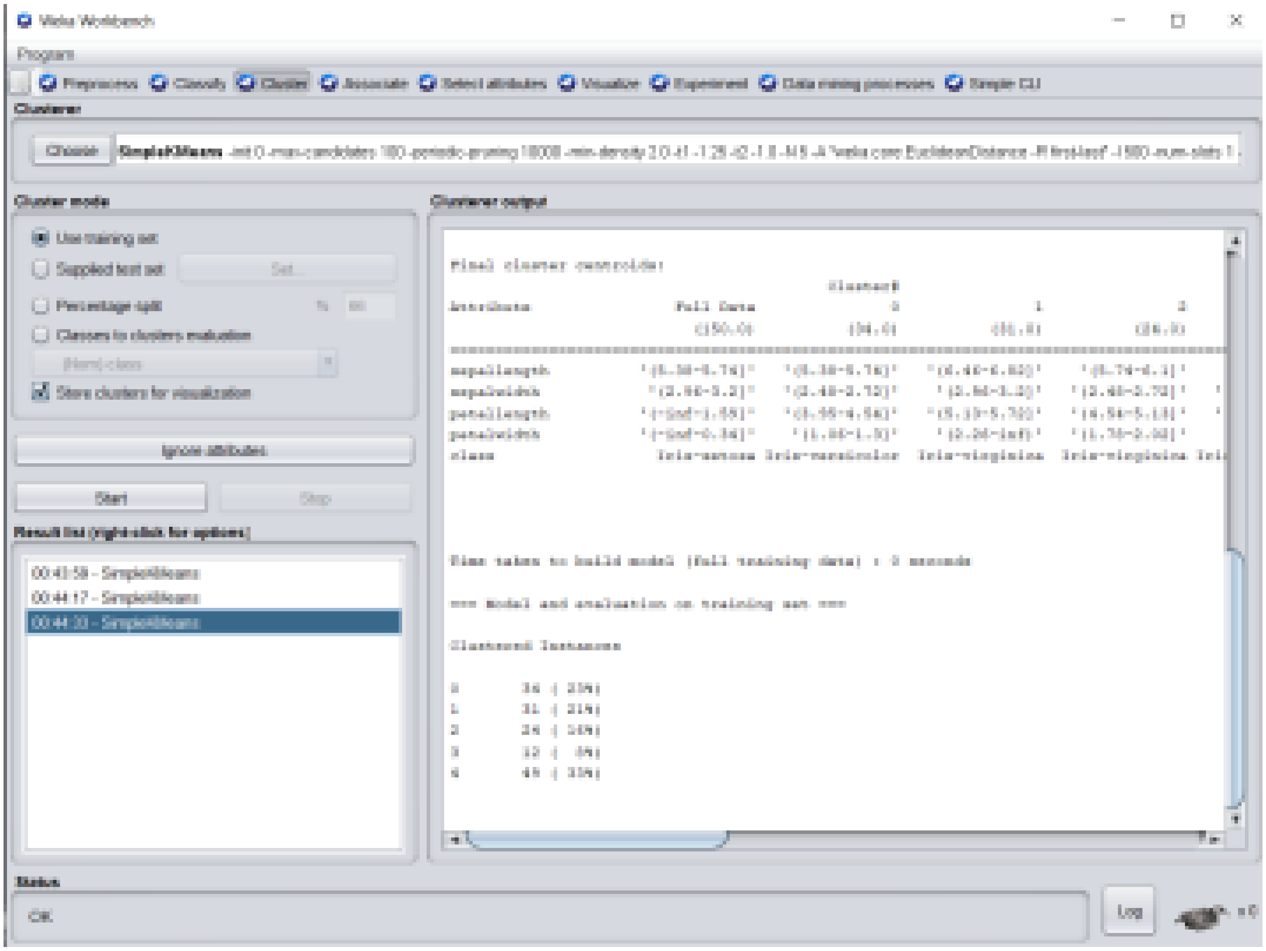
Result list:
08:40:50 - rules.ZeroR
08:41:42 - bayes.NaiveBayes

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

Correctly Classified Instances         143               95.3333 %
Incorrectly Classified Instances         7                4.6667 %
Kappa statistic                          0.93
Mean absolute error                      0.0524
Root mean squared error                  0.1719
Relative absolute error                 11.7846 %
Root relative squared error             36.4371 %
Total Number of Instances              150

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Cl
                 1.000    0.000    1.000      1.000    1.000      1.000    1.000     1.000     Ir
                 0.920    0.030    0.939      0.920    0.929      0.895    0.983     0.962     Ir
                 0.940    0.040    0.922      0.940    0.931      0.896    0.983     0.973     Ir
Weighted Avg.    0.953    0.023    0.953      0.953    0.953      0.930    0.989     0.978

=== Confusion Matrix ===

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

Status: OK

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



SimpleKMeans -init 0 -max-candidates 100 -periodic-pruning 10000 -min-density 2.0 -t1 -1.25 -t2 -1.0 -N 5 -A "weka.core.EuclideanDistance -R first-last" -I 500 -num-slots 1 ...

● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) class
☑ Store clusters for visualization

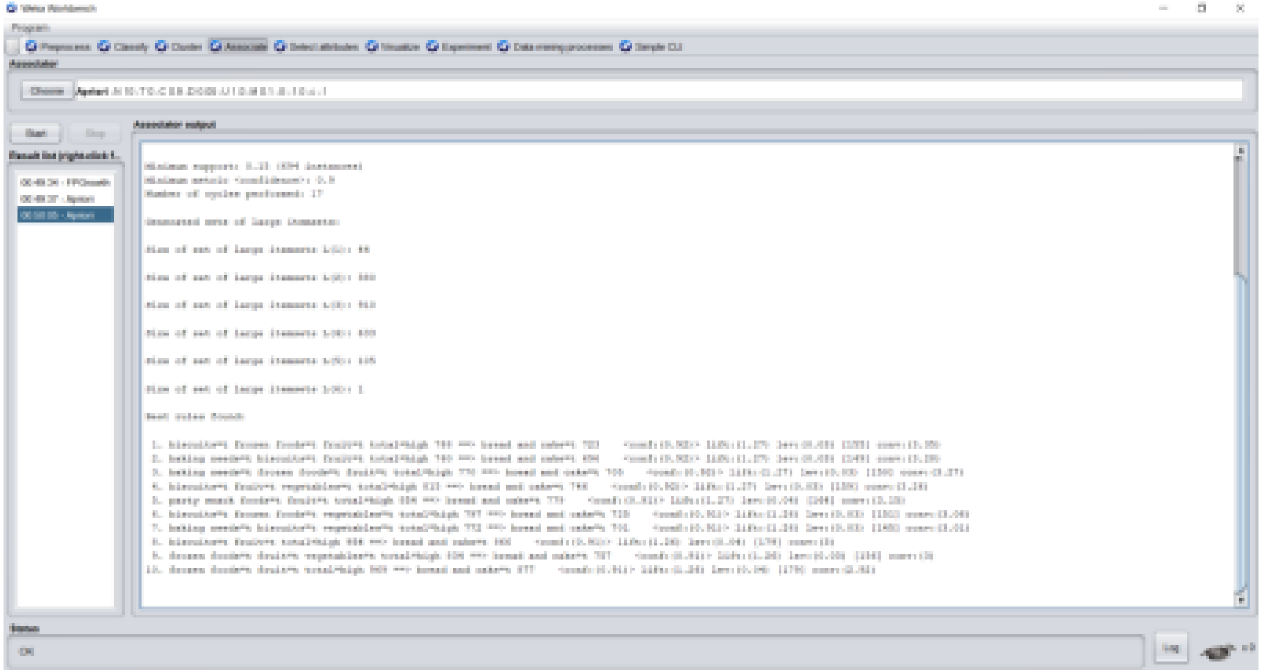
00:43:58 - SimpleKMeans
00:44:17 - SimpleKMeans
00:44:33 - SimpleKMeans

```
Final cluster centroids:
                                            Cluster#
Attribute                   Full Data              0              1              2
                              (150.0)         (34.0)         (31.0)         (26.0)
=====================================================================================
sepallength             '(5.38-5.74]'  '(5.38-5.74]'  '(6.46-6.82]'   '(5.74-6.1]'
sepalwidth               '(2.96-3.2]'  '(2.48-2.72]'   '(2.96-3.2]'  '(2.48-2.72]'  '
petallength             '(-inf-1.59]'  '(3.95-4.54]'  '(5.13-5.72]'  '(4.54-5.13]'  '
petalwidth              '(-inf-0.34]'   '(1.06-1.3]'   '(2.26-inf)'  '(1.78-2.02]'
class                     Iris-setosa Iris-versicolor Iris-virginica Iris-virginica Iri




Time taken to build model (full training data) : 0 seconds

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

Clustered Instances

0       34 ( 23%)
1       31 ( 21%)
2       26 ( 17%)
3       12 (  8%)
4       45 ( 30%)
```

OK

6. **Association rule mining:** Perform apriori algo and show the rules created. The **associate** tab is for automatically finding associations in a dataset. The techniques are often used for market basket analysis type data mining problems and require data where all attributes are categorical.

Here we are using the supermarket dataset and we configure the Apriori algorithm to perform market-basket analysis.



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

```
Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:

 1. biscuits=t frozen foods=t fruit=t total=high 788 ==> bread and cake=t 723    <conf:(0.92)> lift:(1.27) lev:(0.03) [155] conv:(3.35)
 2. baking needs=t biscuits=t fruit=t total=high 760 ==> bread and cake=t 696    <conf:(0.92)> lift:(1.27) lev:(0.03) [149] conv:(3.28)
 3. baking needs=t frozen foods=t fruit=t total=high 770 ==> bread and cake=t 705    <conf:(0.92)> lift:(1.27) lev:(0.03) [150] conv:(3.27)
 4. biscuits=t fruit=t vegetables=t total=high 815 ==> bread and cake=t 746    <conf:(0.92)> lift:(1.27) lev:(0.03) [159] conv:(3.26)
 5. party snack foods=t fruit=t total=high 854 ==> bread and cake=t 779    <conf:(0.91)> lift:(1.27) lev:(0.04) [164] conv:(3.15)
 6. biscuits=t frozen foods=t vegetables=t total=high 797 ==> bread and cake=t 725    <conf:(0.91)> lift:(1.26) lev:(0.03) [151] conv:(3.06)
 7. baking needs=t biscuits=t vegetables=t total=high 772 ==> bread and cake=t 701    <conf:(0.91)> lift:(1.26) lev:(0.03) [145] conv:(3.01)
 8. biscuits=t fruit=t total=high 954 ==> bread and cake=t 866    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(3)
 9. frozen foods=t fruit=t vegetables=t total=high 834 ==> bread and cake=t 757    <conf:(0.91)> lift:(1.26) lev:(0.03) [156] conv:(3)
10. frozen foods=t fruit=t total=high 969 ==> bread and cake=t 877    <conf:(0.91)> lift:(1.26) lev:(0.04) [179] conv:(2.92)
```

CONCLUSION:

We learnt about the Weka tool and how to do data analysis with it. We used 2 different databases: Iris petals and Supermarket.

We tried both the supervised and unsupervised learning algorithms. We can easily visualize with charts how the data transforms when we filter it using different algorithms.

We also used the select attribute to find out which attribute is ranked best for classification. We implemented different clustering and classification algorithms.

In the second database i.e., the supermarket one, we implemented the associate function where we configured the Apriori algorithm to perform market-basket analysis.