EIE3112 Database Systems
Lab 3: Data Mining using Weka on PolyU Virtual Lab Platform - Lab Report

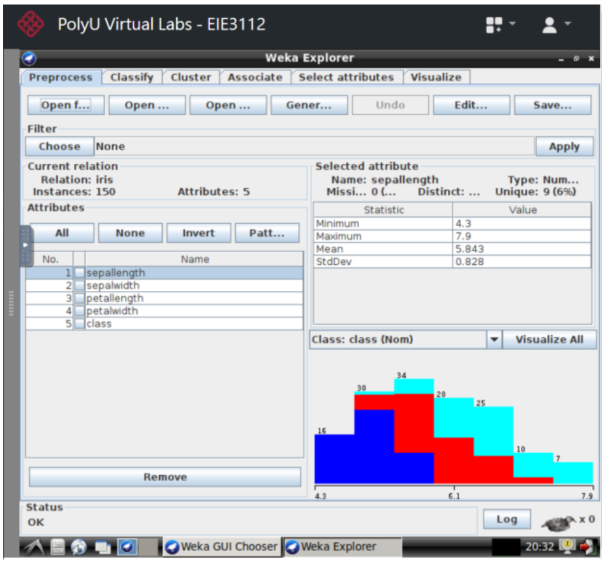
Name: Sze Kin Sang
Student No.: 19066206D

Objectives and Outcomes

- After finishing this lab, you should be able to
- Use Weka to perform data mining
- Understand the differences between classification, cluster analysis, and regression analysis.
- Understand the characteristics of different classifiers and clustering algorithms

Part I: Classification

Procedure:
1)



Filter: None

Current relation
Relation: iris
Instances: 150
Attributes: 5

Selected attribute
Name: sepallength
Type: Num...
Missi... 0 (...
Distinct: ...
Unique: 9 (6%)

Statistic	Value
Minimum	4.3
Maximum	7.9
Mean	5.843
StdDev	0.828

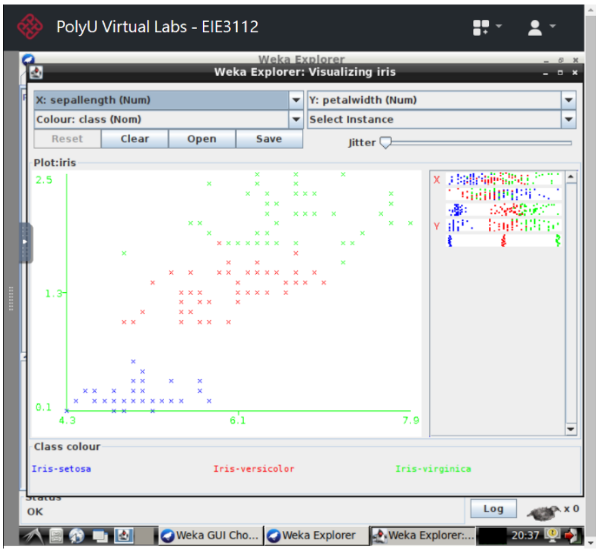
Attributes

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

Class: class (Nom)

Status
OK

Open the WEKA explorer and select the iris.arff file



X: sepallength (Num)

Y: petalwidth (Num)

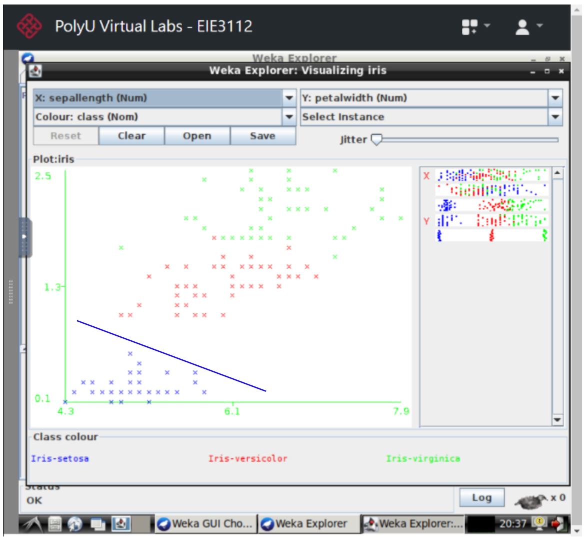
Colour: class (Nom)

Select Instance

Visualizing the data by selecting the suitable attributes

Question 1: Assume that petalwidth and sepallength are used as features. Can linear classifiers such as linear support vector machines classify the training data perfectly? Explain your answer. Include screen capture for your explanation.

According to the graph, it is easy to draw a decision plane to separate class Iris-setosa from other data. However, some of the data of class Iris-versicolor and class Iris-virginica overlap each other, which difficult to draw a decision plane. Thus, linear support vector machines cannot classify the training data perfectly.



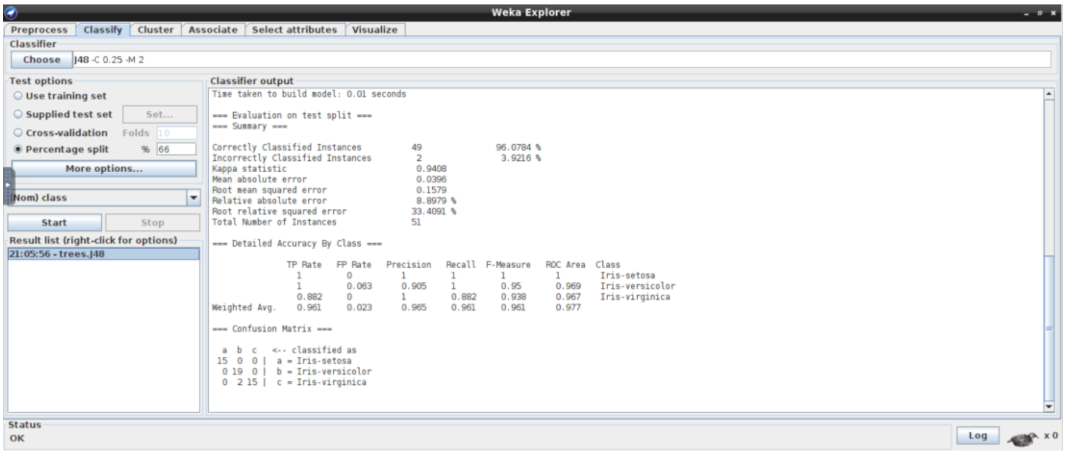
X: sepallength (Num)

Y: petalwidth (Num)

Colour: class (Nom)

Select Instance

2)



Classifier: J48 -C 0.25 -M 2

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

(Nom) class

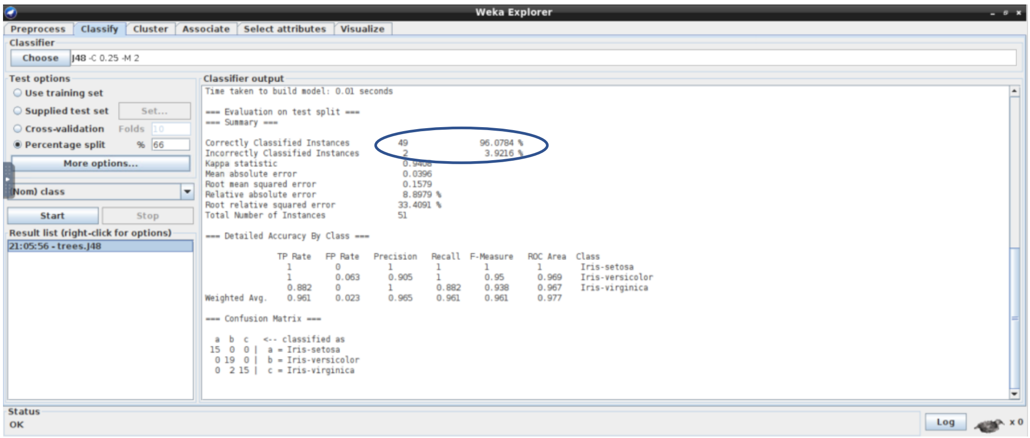
Classifier output

```
Time taken to build model: 0.01 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          49               96.0784 %
Incorrectly Classified Instances         2                3.9216 %
Kappa statistic                          0.9408
Mean absolute error                      0.0396
Root mean squared error                  0.1579
Relative absolute error                  8.8979 %
Root relative squared error             33.4091 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 1         0.063      0.905     1         0.95       0.969    Iris-versicolor
                 0.882     0          1         0.882     0.938      0.967    Iris-virginica
Weighted Avg.    0.961     0.023      0.965     0.961     0.961      0.977

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  2 15 |  c = Iris-virginica
```

Train a decision tree by using J48 program

Question 2: What is the accuracy (correctly classified instances) on the test data? Capture the portion of classifier output that contains the accuracy and circle it. Then put it in your Lab report.

The accuracy on the test data is 96.0784%.



Classifier: J48 -C 0.25 -M 2

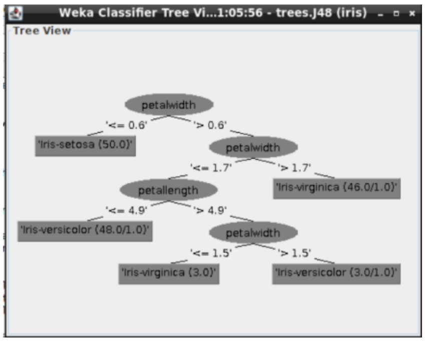
Test options:
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) class

Classifier output

```
Time taken to build model: 0.01 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          49               96.0784 %
Incorrectly Classified Instances         2                3.9216 %
Kappa statistic                          0.9408
Mean absolute error                      0.0396
Root mean squared error                  0.1579
Relative absolute error                  8.8979 %
Root relative squared error             33.4091 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 1         0.063      0.905     1         0.95       0.969    Iris-versicolor
                 0.882     0          1         0.882     0.938      0.967    Iris-virginica
Weighted Avg.    0.961     0.023      0.965     0.961     0.961      0.977

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  2 15 |  c = Iris-virginica
```

3)

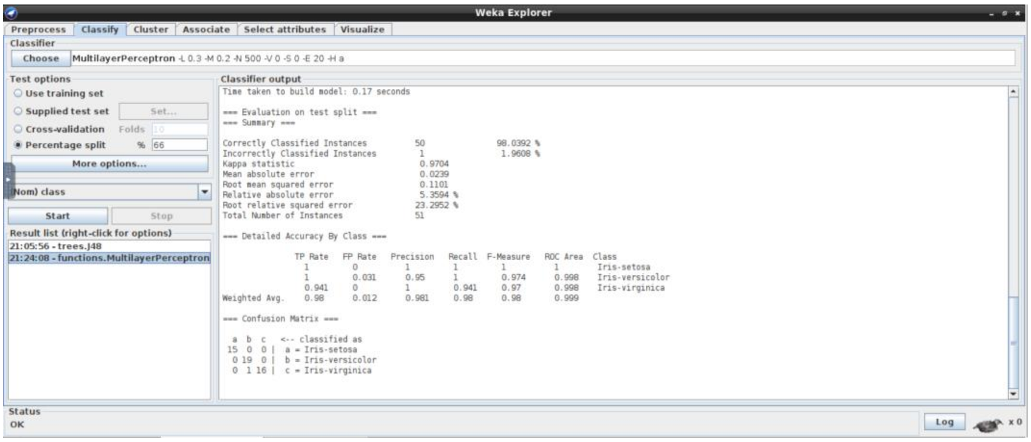


Visualizing the trained decision tree

Question 3: Identify the attribute(s) that is (are) not relevant to the classification task. Explain your answer

Sepallength and sepalwidth are not relevant to the classification task since they didn’t appear in the decision trees, which means the classes can be identify without these two attributes.

4)



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options: ● Percentage split % 66

(Nom) class

Result list (right-click for options)
21:05:56 - trees.J48
21:24:08 - functions.MultilayerPerceptron

Classifier output

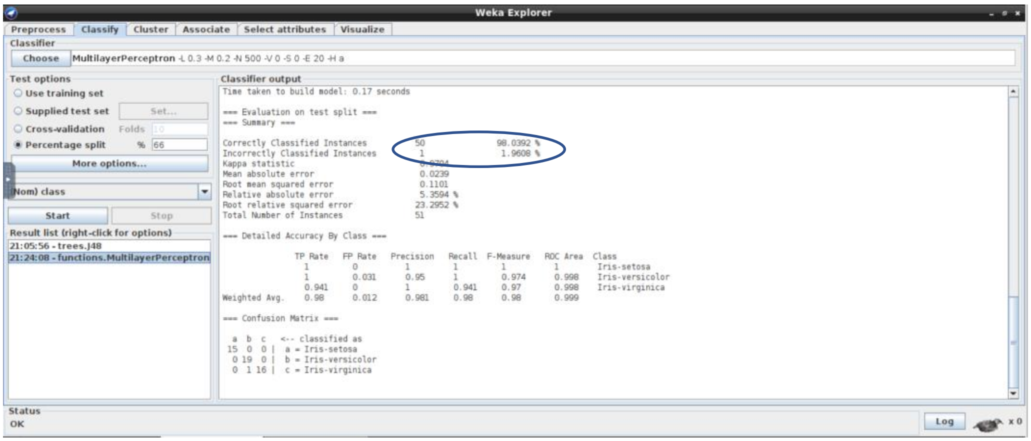
```
Time taken to build model: 0.17 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          50               98.0392 %
Incorrectly Classified Instances         1                1.9608 %
Kappa statistic                          0.9704
Mean absolute error                      0.0239
Root mean squared error                  0.1101
Relative absolute error                  5.3594 %
Root relative squared error             23.2952 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 1         0.031      0.95      1         0.974      0.998    Iris-versicolor
                 0.941     0          1         0.941     0.97       0.998    Iris-virginica
Weighted Avg.    0.98      0.012      0.981     0.98      0.98       0.999

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  1 16 |  c = Iris-virginica
```

Status
OK

Train an artificial neural network by using MultilayerPerceptron program

Question 4: What is the accuracy of MultilayerPerceptron on the iris dataset? Capture the portion of classifier output that contains the accuracy and circle it. Then put it in your Lab report.

The accuracy on the test data is 98.0392%.



Classifier: MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options: ● Percentage split % 66

(Nom) class

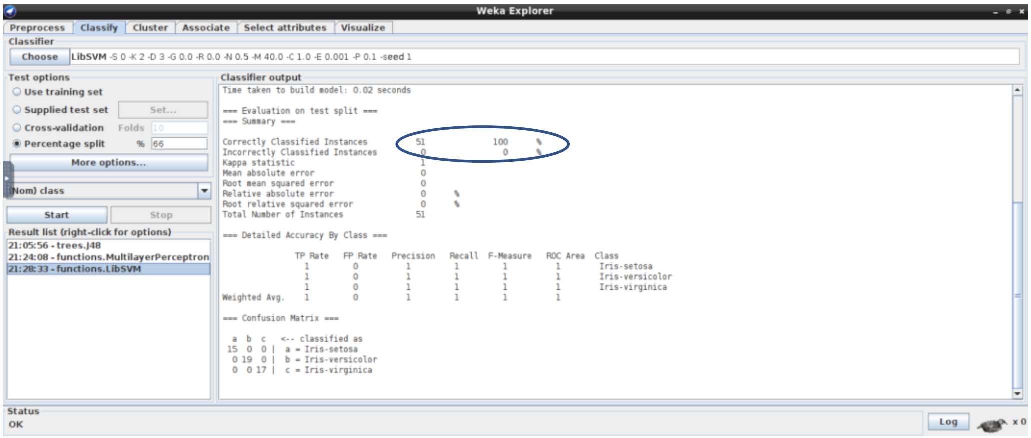
Result list (right-click for options)
21:05:56 - trees.J48
21:24:08 - functions.MultilayerPerceptron

Classifier output

```
Time taken to build model: 0.17 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          50               98.0392 %
Incorrectly Classified Instances         1                1.9608 %
Kappa statistic                          0.9704
Mean absolute error                      0.0239
Root mean squared error                  0.1101
Relative absolute error                  5.3594 %
Root relative squared error             23.2952 %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         0          1         1         1          1        Iris-setosa
                 1         0.031      0.95      1         0.974      0.998    Iris-versicolor
                 0.941     0          1         0.941     0.97       0.998    Iris-virginica
Weighted Avg.    0.98      0.012      0.981     0.98      0.98       0.999

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  1 16 |  c = Iris-virginica
```

Status
OK

5)



Classifier: LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -P 0.1 -seed 1

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

(Nom) class

Result list (right-click for options)
21:05:56 - trees.J48
21:24:08 - functions.MultilayerPerceptron
21:28:33 - functions.LibSVM

Classifier output

```
Time taken to build model: 0.02 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          51              100      %
Incorrectly Classified Instances         0                0      %
Kappa statistic                          1
Mean absolute error                      0
Root mean squared error                  0
Relative absolute error                  0       %
Root relative squared error              0       %
Total Number of Instances               51

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall   F-Measure   ROC Area  Class
                 1         0         1          1         1          1        Iris-setosa
                 1         0         1          1         1          1        Iris-versicolor
                 1         0         1          1         1          1        Iris-virginica
Weighted Avg.    1         0         1          1         1          1

=== Confusion Matrix ===

  a  b  c   <-- classified as
 15  0  0 |  a = Iris-setosa
  0 19  0 |  b = Iris-versicolor
  0  0 17 |  c = Iris-virginica
```

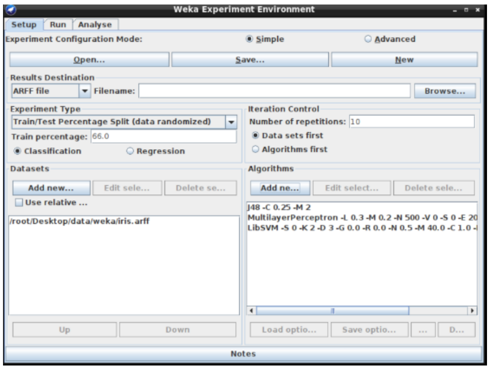
Status
OK

Train a support vector machine by using LibSVM program

Question 5: What is the accuracy of the SVM classifier on the iris dataset? Capture the portion of classifier output that contains the accuracy and circle it. Then put it in your Lab report.

The accuracy on the test data is 100%.

6)



Experiment Configuration Mode: ● Simple ○ Advanced

Results Destination: ARFF file Filename:

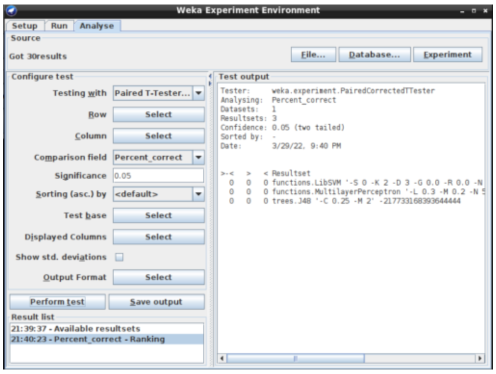
Experiment Type: Train/Test Percentage Split (data randomized)
Train percentage: 66.0
● Classification ○ Regression

Iteration Control: Number of repetitions: 10
● Data sets first
○ Algorithms first

Datasets: □ Use relative ...
/root/Desktop/data/weka/iris.arff

Algorithms:
J48 -C 0.25 -M 2
MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20
LibSVM -S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -

Designing an experiment using requested dataset and algorithms



Source: Got 30results

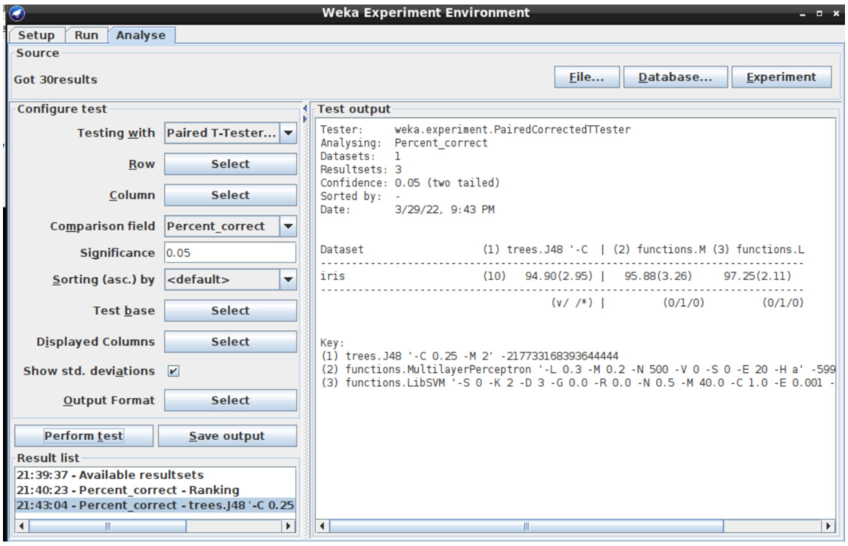
Configure test
Testing with: Paired T-Tester...
Row: Select
Column: Select
Comparison field: Percent_correct
Significance: 0.05
Sorting (asc.) by: <default>
Test base: Select
Displayed Columns: Select
Show std. deviations □
Output Format: Select

Result list
21:39:37 - Available resultsets
21:40:23 - Percent_correct - Ranking

Test output

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       3/29/22, 9:40 PM


>-<   >   < Resultset
  0   0   0 functions.LibSVM '-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N
  0   0   0 functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 5
  0   0   0 trees.J48 '-C 0.25 -M 2' -217733168393644444
```

Analyse the result of experiment to rank the algorithms



Source

Got 30results

Configure test

Testing with: Paired T-Tester...
Comparison field: Percent_correct
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations ☑

Result list

21:39:37 - Available resultsets
21:40:23 - Percent_correct - Ranking
21:43:04 - Percent_correct - trees.J48 '-C 0.25

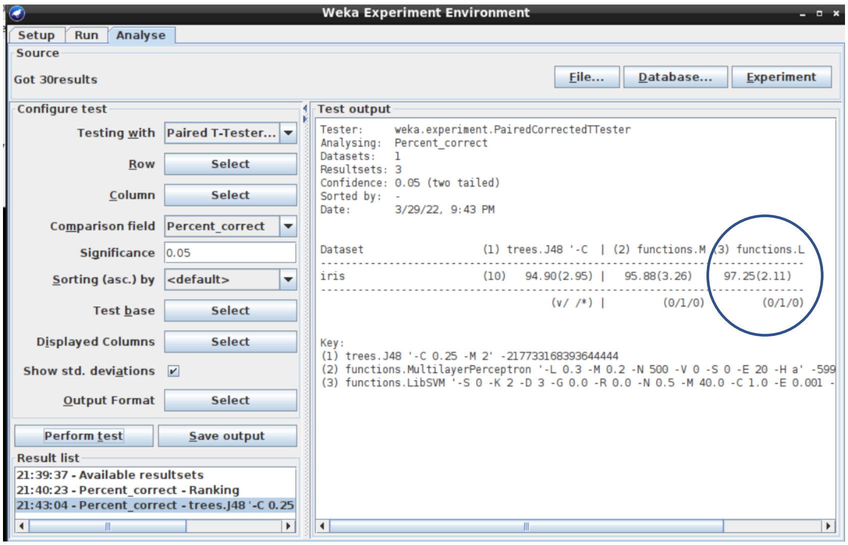
Test output

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       3/29/22, 9:43 PM


Dataset                   (1) trees.J48 '-C  | (2) functions.M (3) functions.L
------------------------------------------------------------------------------
iris                      (10)   94.90(2.95) |   95.88(3.26)     97.25(2.11)
------------------------------------------------------------------------------
                                     (v/ /*) |         (0/1/0)         (0/1/0)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -599
(3) functions.LibSVM '-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -
```

Analyse the result of experiment to show the accuracy of algorithms

Question 6: Which classifier is more accurate in classifying data in the iris dataset? What is the accuracy of the best classifier? Capture the portion of Test output that contains the accuracy of all three classifiers. Circle the name of the best classifier together with its accuracy. Then put it in your Lab report.

According to the graph, since LibSVM has the highest accuracy than other two classifiers, which is 97.25%, it is the best classifier for the iris dataset.



Source

Got 30results

Configure test

Testing with: Paired T-Tester...
Comparison field: Percent_correct
Significance: 0.05
Sorting (asc.) by: <default>
Show std. deviations ☑

Result list

21:39:37 - Available resultsets
21:40:23 - Percent_correct - Ranking
21:43:04 - Percent_correct - trees.J48 '-C 0.25

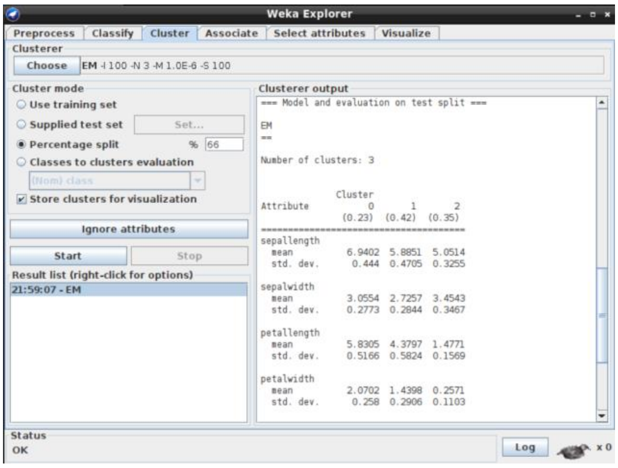
Test output

```
Tester:     weka.experiment.PairedCorrectedTTester
Analysing:  Percent_correct
Datasets:   1
Resultsets: 3
Confidence: 0.05 (two tailed)
Sorted by:  -
Date:       3/29/22, 9:43 PM


Dataset                   (1) trees.J48 '-C  | (2) functions.M (3) functions.L
------------------------------------------------------------------------------
iris                      (10)   94.90(2.95) |   95.88(3.26)     97.25(2.11)
------------------------------------------------------------------------------
                                     (v/ /*) |         (0/1/0)         (0/1/0)


Key:
(1) trees.J48 '-C 0.25 -M 2' -217733168393644444
(2) functions.MultilayerPerceptron '-L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a' -599
(3) functions.LibSVM '-S 0 -K 2 -D 3 -G 0.0 -R 0.0 -N 0.5 -M 40.0 -C 1.0 -E 0.001 -
```

Part II: Clustering

Procedure:

1)



Clusterer

EM -I 100 -N 3 -M 1.0E-6 -S 100

Cluster mode

○ Use training set

○ Supplied test set

● Percentage split % 66

○ Classes to clusters evaluation

(Nom) class

☑ Store clusters for visualization

Result list (right-click for options)

21:59:07 - EM

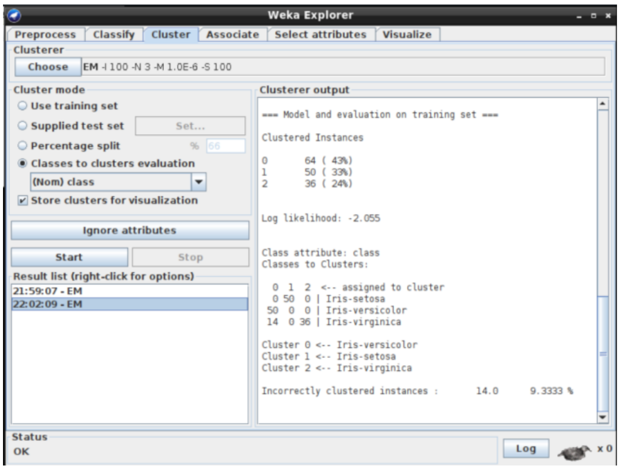
Clusterer output

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

EM
==

Number of clusters: 3


                 Cluster
Attribute             0        1        2
                 (0.23)   (0.42)   (0.35)
=========================================
sepallength
  mean           6.9402   5.8851   5.0514
  std. dev.       0.444   0.4705   0.3255

sepalwidth
  mean           3.0554   2.7257   3.4543
  std. dev.      0.2773   0.2844   0.3467

petallength
  mean           5.8305   4.3797   1.4771
  std. dev.      0.5166   0.5824   0.1569

petalwidth
  mean           2.0702   1.4398   0.2571
  std. dev.       0.258   0.2906   0.1103
```

Status

OK

Designing a cluster analysation with given parameter using EM



Clusterer

EM -I 100 -N 3 -M 1.0E-6 -S 100

Cluster mode

○ Use training set

○ Supplied test set

○ Percentage split % 66

● Classes to clusters evaluation

(Nom) class

☑ Store clusters for visualization

Result list (right-click for options)

21:59:07 - EM

22:02:09 - EM

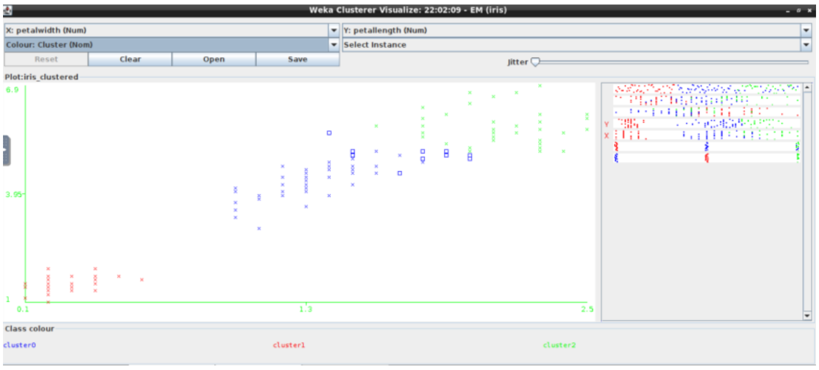
Clusterer output

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

Clustered Instances

0       64 ( 43%)
1       50 ( 33%)
2       36 ( 24%)


Log likelihood: -2.055


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 50  0  0 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :      14.0      9.3333 %
```

Status

OK

Evaluting the cluster data by assigning classes to it

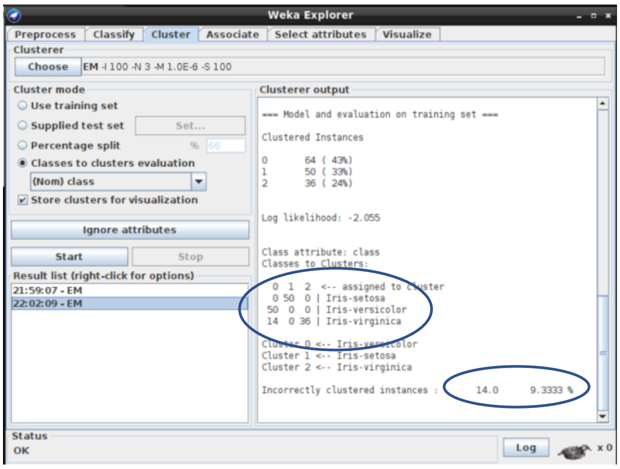


Visualizing the cluster assignment and incorrectly clustered instances appear

Question 7: Given that each class comprises 50 samples, which of the two classes contain the largest number of confusable data? Explain your answer. Note that if the cluster assignment is perfect, you should have seen the following results in the "Cluster output" window:

```
0    1    2    <-- assigned to cluster
0    50   0    | Iris-setosa
50   0    0    | Iris-versicolor
0    0    50   | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica
```

Class Iris-setosa and Iris-virginica since according to the graph, there are 14 data records are confusable which they belong to Iris-setosa or Iris-virginica.



Clusterer: EM -I 100 -N 3 -M 1.0E-6 -S 100

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

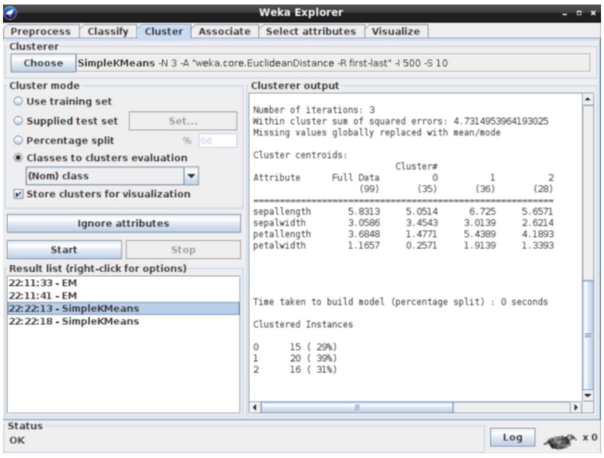
Result list (right-click for options)
21:59:07 - EM
22:02:09 - EM

Clusterer output

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

Clustered Instances

0       64 ( 43%)
1       50 ( 33%)
2       36 ( 24%)


Log likelihood: -2.055


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 50  0  0 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :       14.0       9.3333 %
```

Status: OK

2)



Clusterer: SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

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

Result list (right-click for options)
22:11:33 - EM
22:11:41 - EM
22:22:13 - SimpleKMeans
22:22:18 - SimpleKMeans

Clusterer output

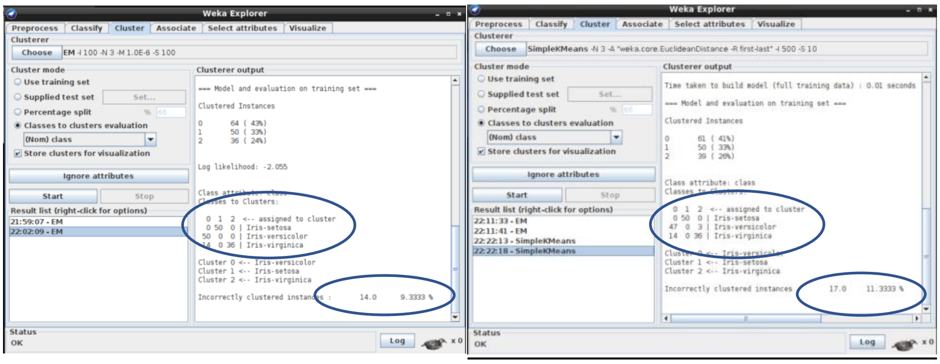
```
Number of iterations: 3
Within cluster sum of squared errors: 4.7314953964193025
Missing values globally replaced with mean/mode

Cluster centroids:
                              Cluster#
Attribute      Full Data          0          1          2
                    (99)       (35)       (36)       (28)
==========================================================
sepallength       5.8313     5.0514      6.725     5.6571
sepalwidth        3.0586     3.4543     3.0139     2.6214
petallength       3.6848     1.4771     5.4389     4.1893
petalwidth        1.1657     0.2571     1.9139     1.3393




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

Clustered Instances

0       15 ( 29%)
1       20 ( 39%)
2       16 ( 31%)
```

Status: OK

Designing a cluster analysation with given parameter using K-Means

Question 8: How many samples are wrongly clustered by K-means? How many samples are wrongly clustered by EM? Is K-means better or poorer than EM in clustering the iris data? Capture the portion of Clusterer output that contains the number of wrongly clustered samples by K-means and by EM. Circle the number and put it in your Lab report.

There are 17 incorrectly clustered instances using K-means while there are only 14 incorrectly clustered instances using EM. Therefore, for this iris data, K-mean perform poorer than EM in clustering data.



Clusterer: EM -I 100 -N 3 -M 1.0E-6 -S 100

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

Result list (right-click for options)
21:59:07 - EM
22:02:09 - EM

Clusterer output

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

Clustered Instances

0       64 ( 43%)
1       50 ( 33%)
2       36 ( 24%)


Log likelihood: -2.055


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 50  0  0 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances :      14.0      9.3333 %
```

Clusterer: SimpleKMeans -N 3 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

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

Result list (right-click for options)
22:11:33 - EM
22:11:41 - EM
22:22:13 - SimpleKMeans
22:22:18 - SimpleKMeans

Clusterer output

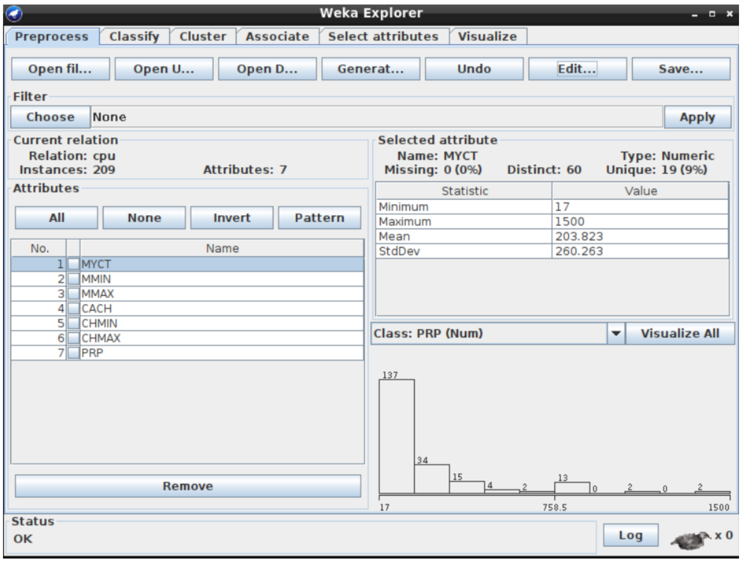
```
Time taken to build model (full training data) : 0.01 seconds

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

Clustered Instances

0       61 ( 41%)
1       50 ( 33%)
2       39 ( 26%)


Class attribute: class
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0 50  0 | Iris-setosa
 47  0  3 | Iris-versicolor
 14  0 36 | Iris-virginica

Cluster 0 <-- Iris-versicolor
Cluster 1 <-- Iris-setosa
Cluster 2 <-- Iris-virginica

Incorrectly clustered instances        17.0      11.3333 %
```

Result generated by using (i) EM (ii) K-Means

Part III: Regression

Procedure:

1)



Filter: None

Current relation
Relation: cpu
Instances: 209 Attributes: 7

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

Statistic	Value
Minimum	17
Maximum	1500
Mean	203.823
StdDev	260.263

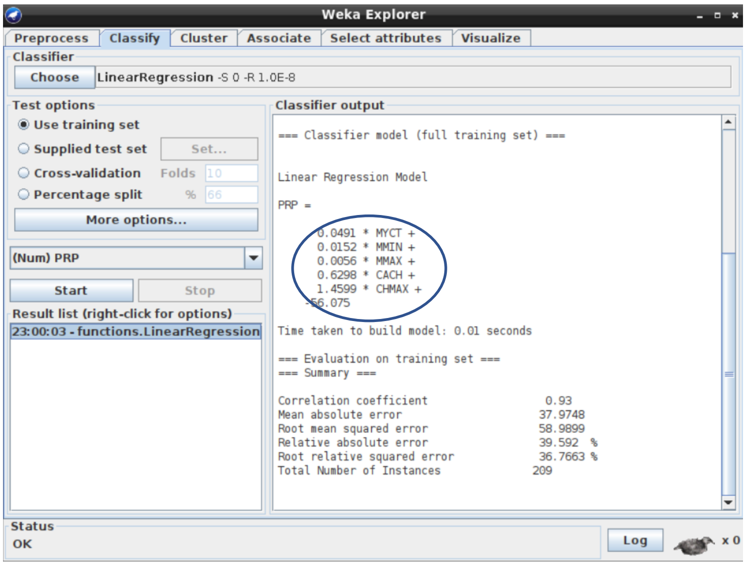
Attributes

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

Class: PRP (Num)

Status
OK

Open the WEKA explorer and select the cpu.arff file



Classifier: LinearRegression -S 0 -R 1.0E-8

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

(Num) PRP

23:00:03 - functions.LinearRegression

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


Linear Regression Model

PRP =

      0.0491 * MYCT +
      0.0152 * MMIN +
      0.0056 * MMAX +
      0.6298 * CACH +
      1.4599 * CHMAX +
    -56.075

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===
=== Summary ===

Correlation coefficient                  0.93
Mean absolute error                     37.9748
Root mean squared error                 58.9899
Relative absolute error                 39.592  %
Root relative squared error             36.7663 %
Total Number of Instances              209
```

Status
OK

Design a linear regression using given parameter

Question 9: Which of the attributes does not affect the performance of the CPU? Explain your answer.

Attribute CHMIN doesn't affect the performance of the CPU since it doesn't appear in the linear regression model. This indicates the rating of performance doesn't depend on the CHMIN.