**Week – 5 Demonstrate performing classification on data sets**

**Load each dataset into Weka and run 1d3, J48 classification algorithm.**

**Study the classifier output. Compute entropy values, Kappa statistic.**

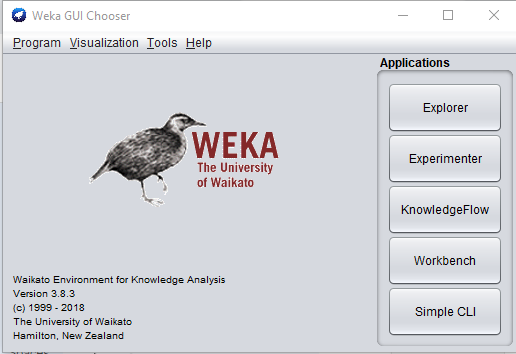
**Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.**

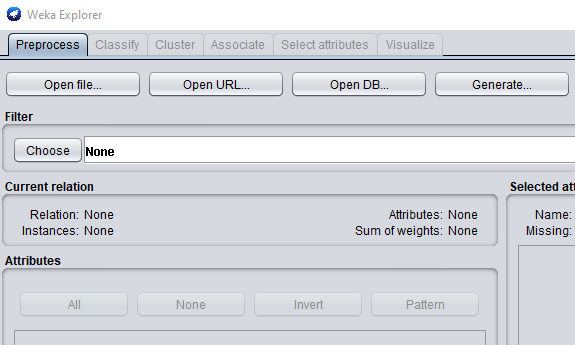
**Load each dataset into Weka and perform Naïve-bayes classification and k-Nearest Neighbour classification.**

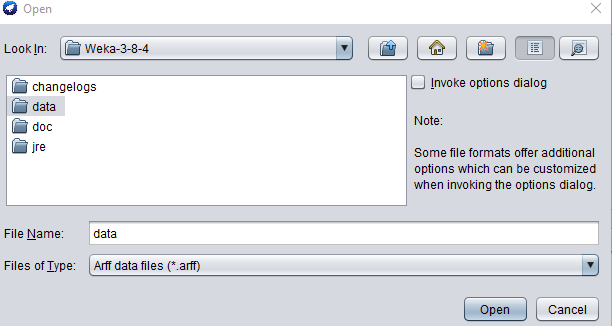
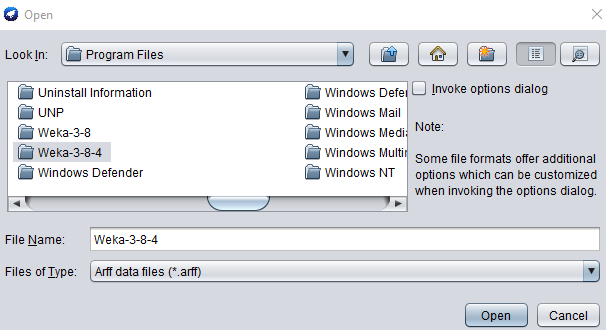
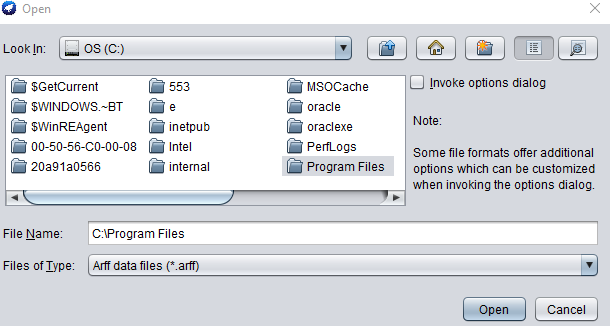
**Interpret the results obtained.**

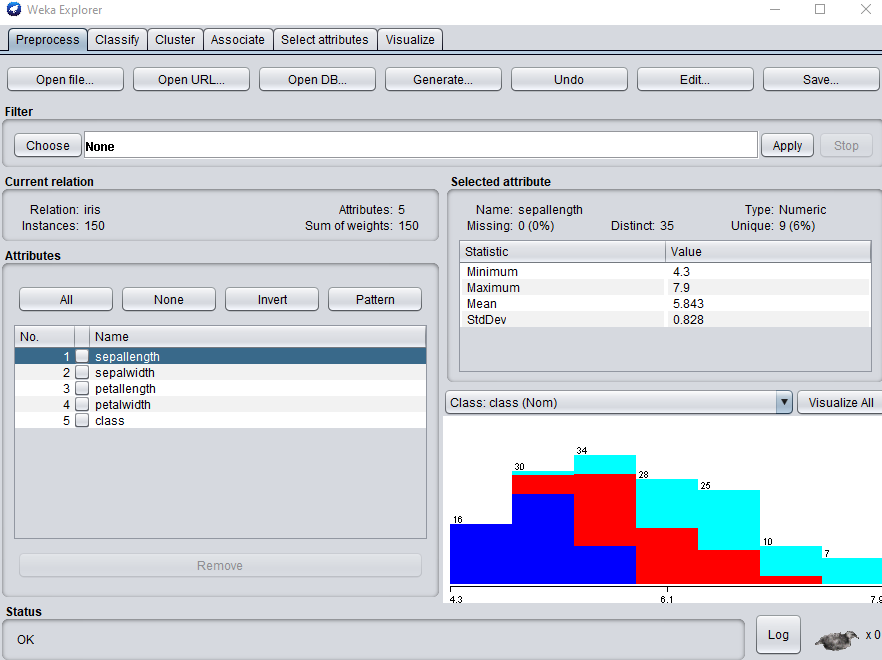
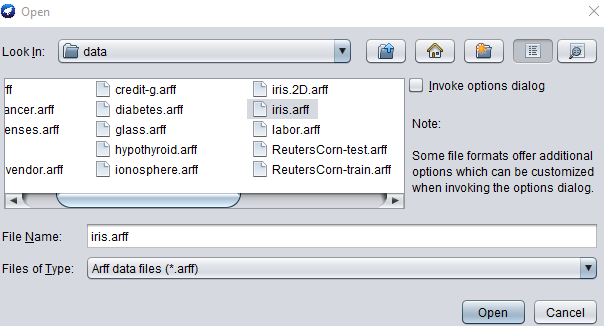
**Plot RoC Curves**

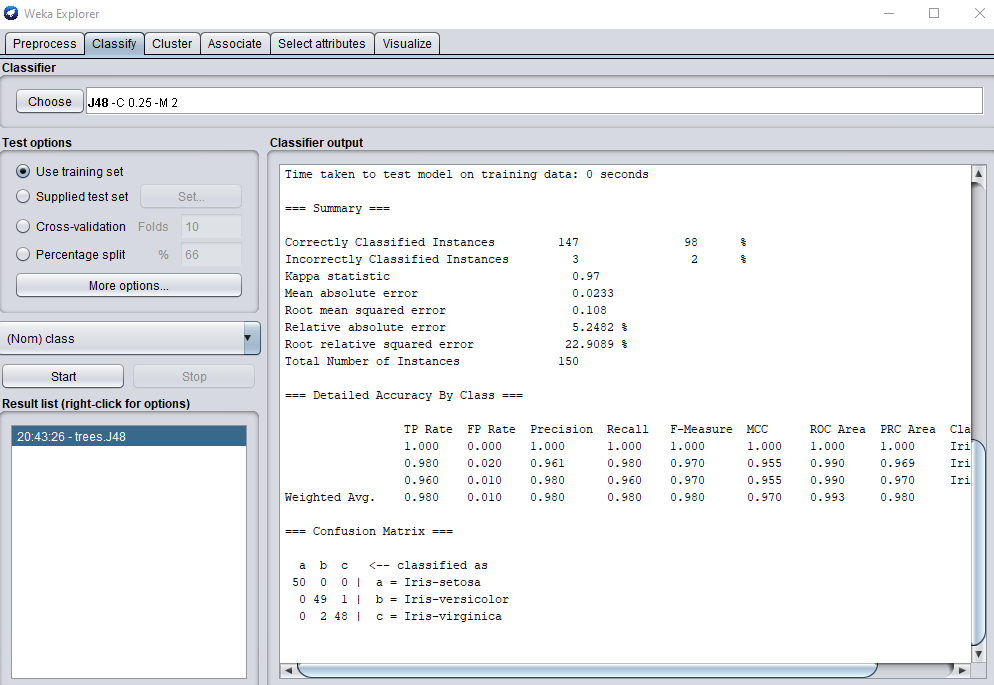
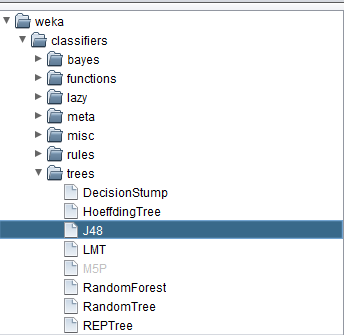
**Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.**

GO TO EXPLORER





GO TO CLASSIFY AND CHOOSE

=== Run information ===

Scheme: weka.classifiers.trees.J48 -C 0.25 -M 2

Relation: iris

Instances: 150

Attributes: 5

sepallength

sepalwidth

petallength

petalwidth

class

Test mode: evaluate on training data

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

J48 pruned tree

------------------

petalwidth <= 0.6: Iris-setosa (50.0)

petalwidth > 0.6

| petalwidth <= 1.7

| | petallength <= 4.9: Iris-versicolor (48.0/1.0)

| | petallength > 4.9

| | | petalwidth <= 1.5: Iris-virginica (3.0)

| | | petalwidth > 1.5: Iris-versicolor (3.0/1.0)

| petalwidth > 1.7: Iris-virginica (46.0/1.0)

Number of Leaves : 5

Size of the tree : 9

Time taken to build model: 0.01 seconds

=== Evaluation on training set ===

Time taken to test model on training data: 0 seconds

=== Summary ===

Correctly Classified Instances 147 98 %

Incorrectly Classified Instances 3 2 %

Kappa statistic 0.97

Mean absolute error 0.0233

Root mean squared error 0.108

Relative absolute error 5.2482 %

Root relative squared error 22.9089 %

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

0.980 0.020 0.961 0.980 0.970 0.955 0.990 0.969 Iris-versicolor

0.960 0.010 0.980 0.960 0.970 0.955 0.990 0.970 Iris-virginica

Weighted Avg. 0.980 0.010 0.980 0.980 0.980 0.970 0.993 0.980

=== Confusion Matrix ===

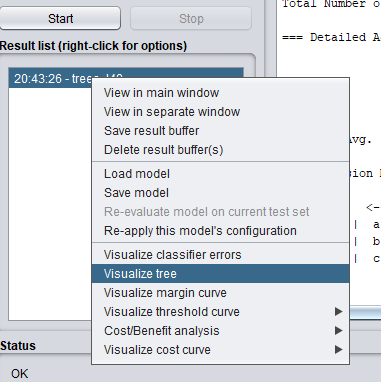
a b c <-- classified as

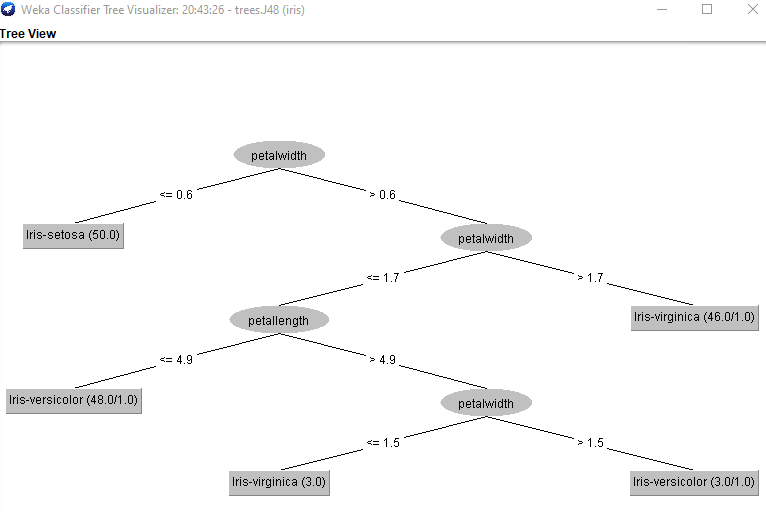
50 0 0 | a = Iris-setosa

0 49 1 | b = Iris-versicolor

0 2 48 | c = Iris-virginica

RIGHT CLICK ON THE TREE OPTION





The kappa statistic, which takes into account chance agreement, is defined as **(observed agreement−expected agreement)/(1−expected agreement)**.

Mean Absolute Error calculates the average difference between the calculated values and actual values. It is also known as scale-dependent accuracy as it calculates error in observations taken on the same scale. It is used as evaluation metrics for regression models in machine learning. It calculates errors between actual values and values predicted by the model. It is used to predict the accuracy of the machine learning model.

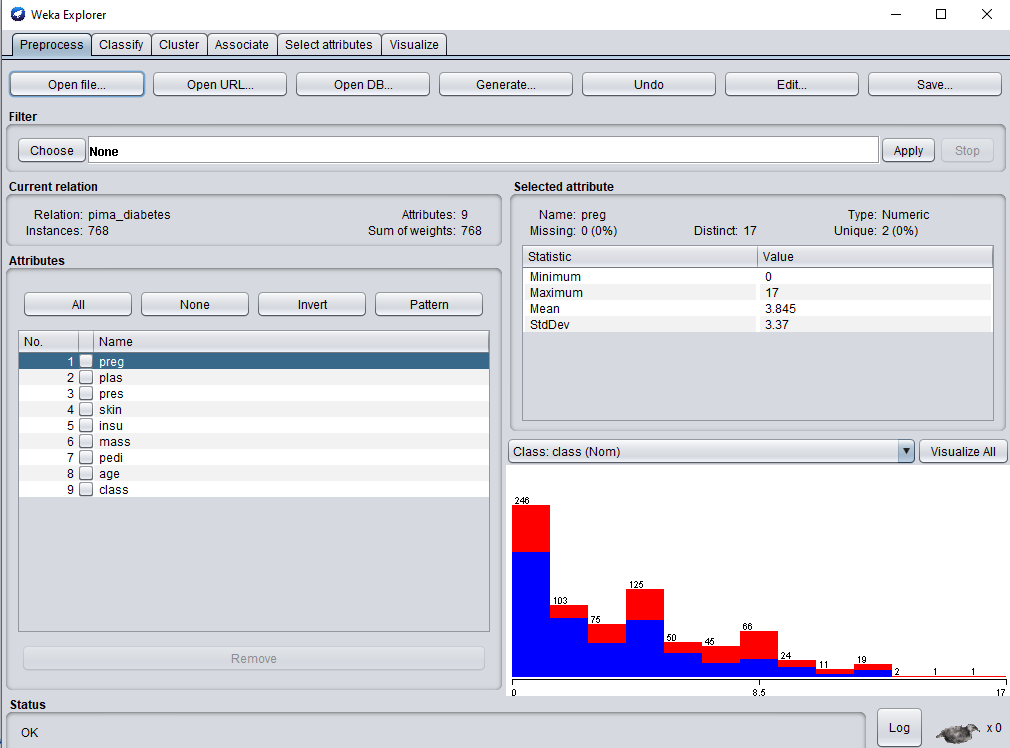
**Formula:**

*Mean Absolute Error = (1/n) \* ∑|yi – xi|*

*where,*

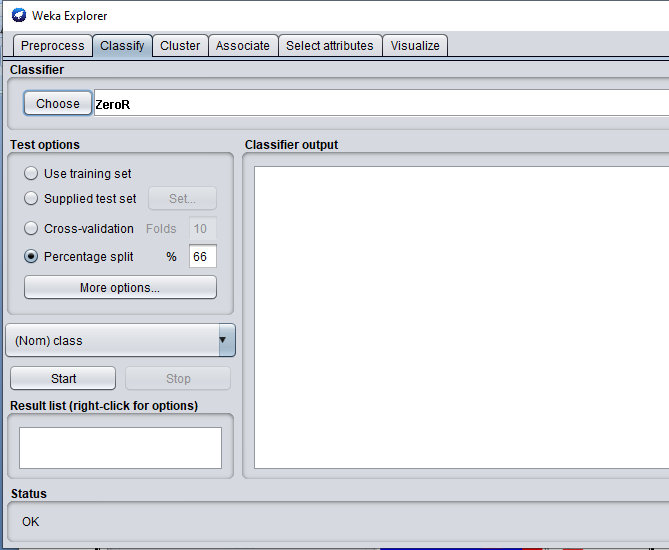
* *Σ: Greek symbol for summation*
* *yi: Actual value for the ith observation*
* *xi: Calculated value for the ith observation*
* *n: Total number of observations*

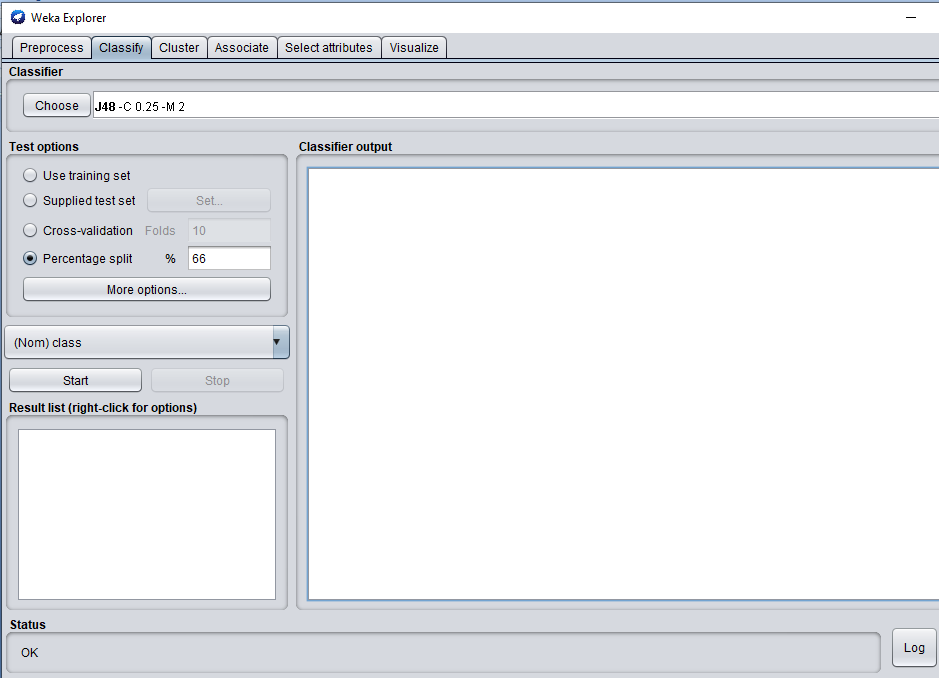
load DIABETES DATA SET

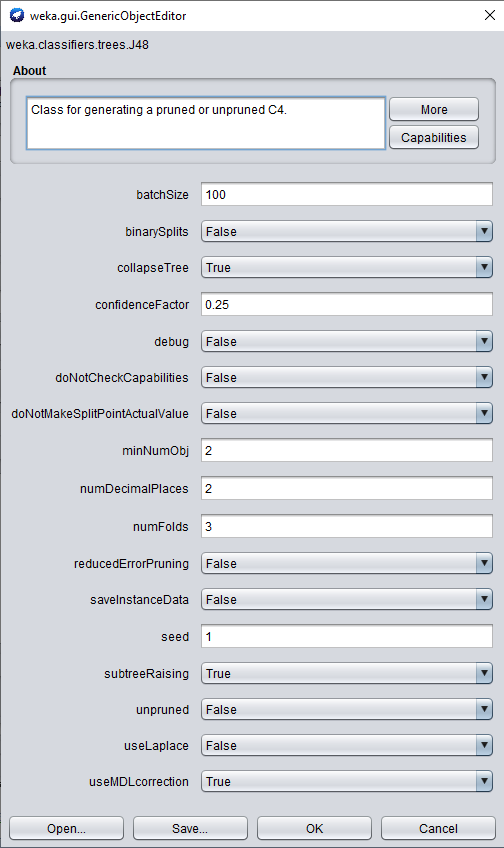


Unique attributes are not required for classification

Click on classify





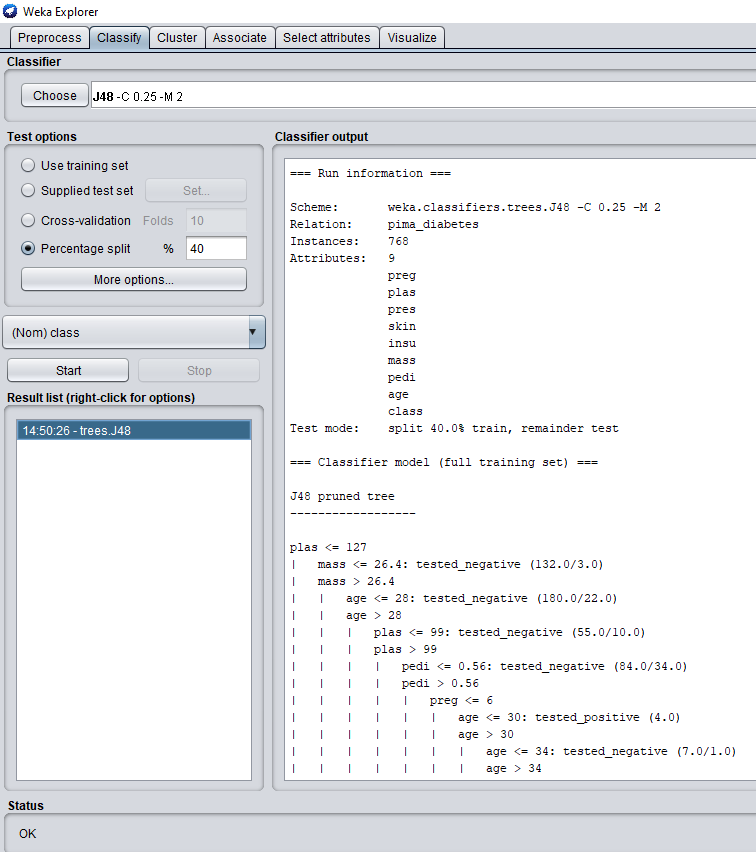


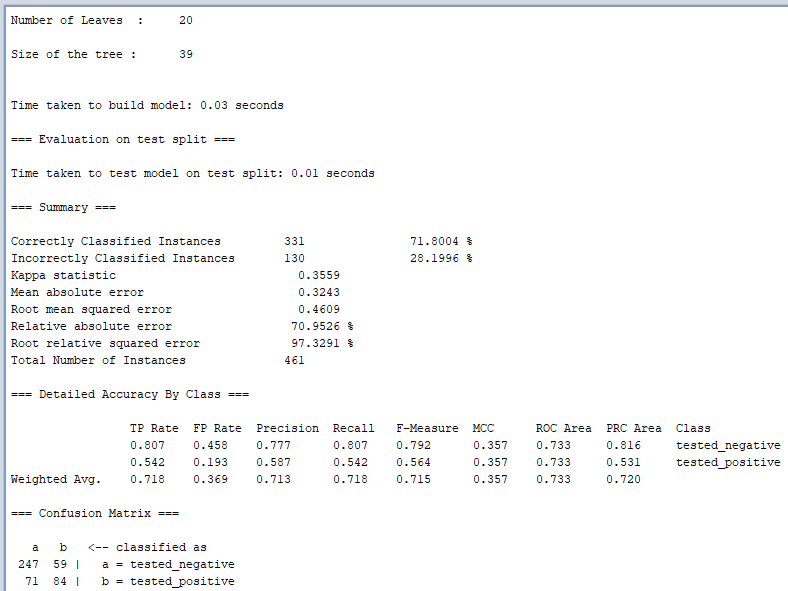
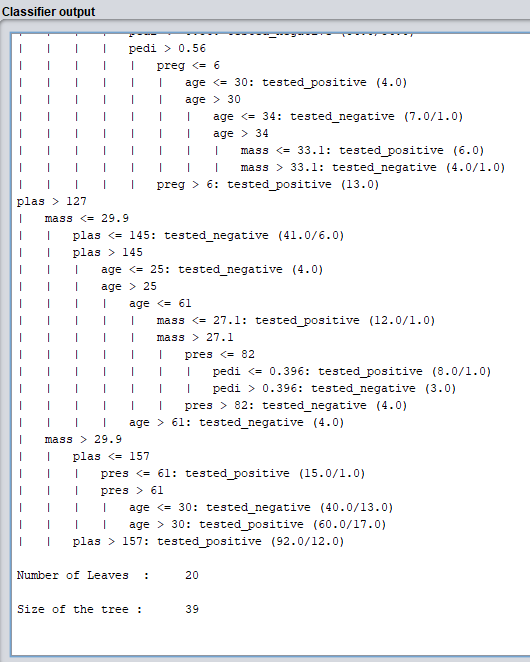
These are the parameters to change before going to classification

Percentage split: 40%

If you have 10 records ,out of that 6 is used for training data set and remaining for test data set

Click on start





Kappa statistic:

Cohen’s kappa statistic measures interrater reliability (sometimes called interobserver agreement). Interrater reliability, or precision, happens when your data raters (or collectors) give the same score to the same data item.

This statistic should only be calculated when:

Two raters each rate one trial on each sample, *or*.

One rater rates two trials on each sample.

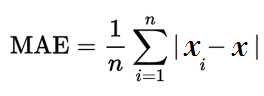
p0 - pe 1 - p0

k=----------- = 1- ----------

1 - pe 1 – pe

Mean Absolute Error

The Mean Absolute Error(MAE) is the [average](https://www.calculushowto.com/average-value-of-a-function/#def) of all absolute errors.

The formula is:  
 [](https://www.statisticshowto.com/wp-content/uploads/2016/10/MAE.png)

Where:

n = the number of errors,

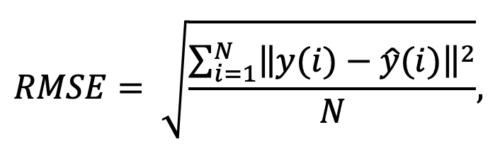
Σ = [summation symbol](https://calculushowto.com/what-is-sigma-summation-notation/) (which means “add them all up”),

|xi – x| = the absolute errors.

Root mean squared error:

Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

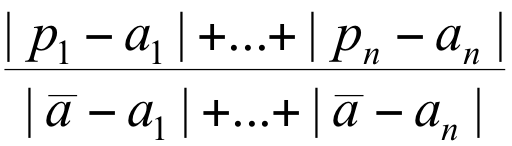
Root mean square error can be expressed as



where N is the number of data points, y(i) is the i-th measurement, and y ̂(i) is its corresponding prediction.

Relative absolute error:

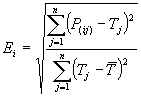
**It** is a way to measure the performance of a predictive model. It’s primarily used in machine learning, data mining, and operations management. RAE is not to be confused with [relative error](https://www.statisticshowto.com/relative-error/), which is a general measure of precision or accuracy for instruments like clocks, rulers, or scales.



**Root relative squared error:**

The Root Relative Squared Error (RRSE) is defined as the square root of the sum of squared errors of a predictive model normalized by the sum of squared errors of a simple model.

the root relative squared error *Ei* of an individual model *i* is evaluated by the equation:



where *P*(*ij*) is the value predicted by the individual model *i* for record *j* (out of *n* records); *Tj* is the target value for record *j*; andhttps://www.gepsoft.com/gxpt4kb/Chapter09/Section1/SS07.h13.gifis given by the formula:

https://www.gepsoft.com/gxpt4kb/Chapter09/Section1/SS07.h14.gif

For a perfect fit, the numerator is equal to 0 and *Ei* = 0. So, the *Ei* index ranges from 0 to infinity, with 0 corresponding to the ideal.

, where:

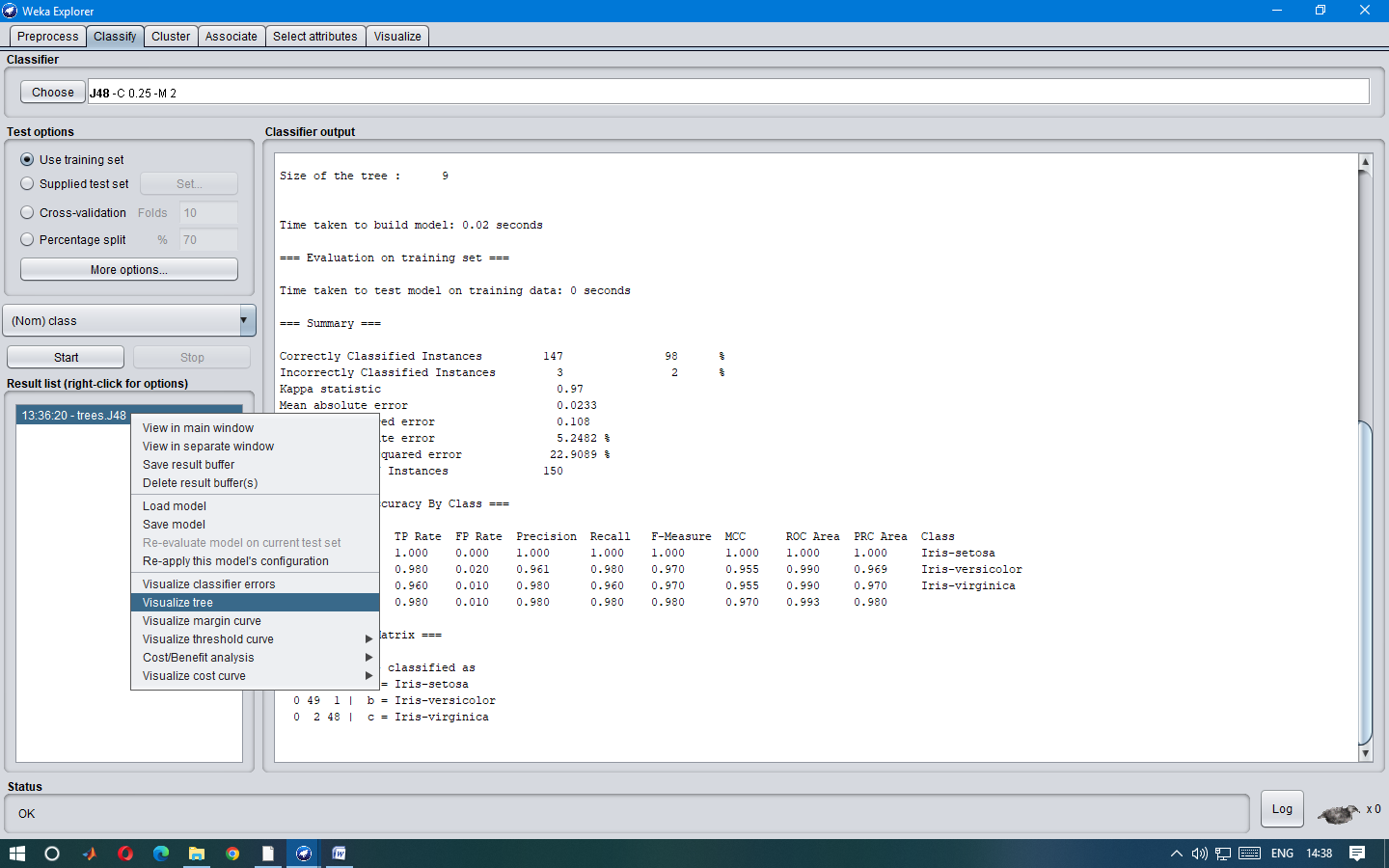
*n*: represents the number of observations

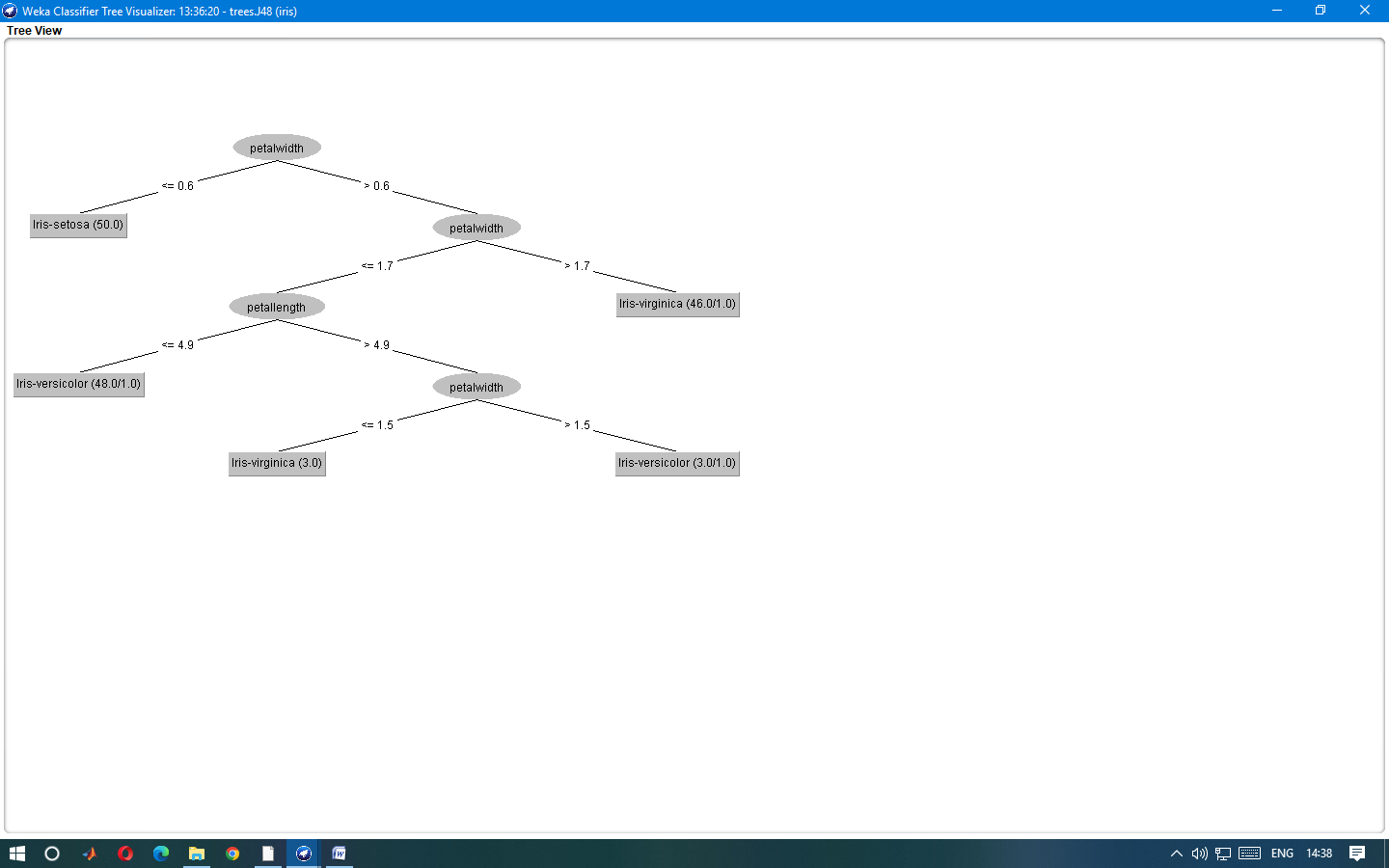
*yi*: represents the realized value

*ŷ*i: represents the predicted value

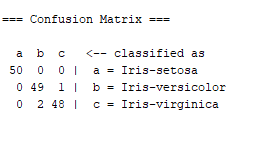
ȳ: represents the average of the realized values

5. Extract if-then rules from the decision tree generated by the classifier.





6. Observe the confusion matrix.



Precision: Appropriate when minimizing false positives is the focus.

Recall: Appropriate when minimizing false negatives is the focus.

TP Rate: rate of true positives (instances correctly classified as a given class)

FP Rate: rate of false positives (instances falsely classified as a given class)

F measure is :

F-Measure = (2 \* Precision \* Recall) / (Precision + Recall)

MCC : it is used in machine learning as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes

ROC( Receiver Operating Characteristics) area measurement: One of the most important values output by Weka. They give you an idea of how the classifiers are performing in general

PRC( Precision Recall) area :

Precision-recall curve. A plot of precision (= PPV) vs. recall (= sensitivity) for all potential cut-offs for a test.