LAB PROGRAM - 3

Consider the data set

No.	eid Numeric	ename Nominal	salary Numeric	exp Numeric	address Nominal
1	101.0	raj	10000.0	4.0	pdtr
2	102.0	ramu	15000.0	5.0	pdtr
3	103.0	anil	12000.0	3.0	kdp
4	104.0	sunil	13000.0	3.0	kdp
5	105.0	rajiv	16000.0	6.0	kdp
6	106.0	sunitha	15000.0	5.0	nlr
7	107.0	kavitha	12000.0	3.0	nlr
8	108.0	suresh	11000.0	5.0	gtr
9	109.0	ravi	12000.0	3.0	gtr
10	110.0	ramana	11000.0	5.0	gtr
11	111.0	ram	12000.0	3.0	kdp
12	112.0	kavya	13000.0	4.0	kdp
13	113.0	navya	14000.0	5.0	kdp

Use the data sources, like ARFF, XML ARFF files. Do the following

i) Classify , Invoke MultiLayerPerception

ii) Build neural network GUI as below

a) Beginning the process of editing the network to add a second hidden layer

b) The finished network with two hidden layers

iii) Apply Lazy classifier, multi instance classifier

iv) Apply any MetaLearning Algorithm

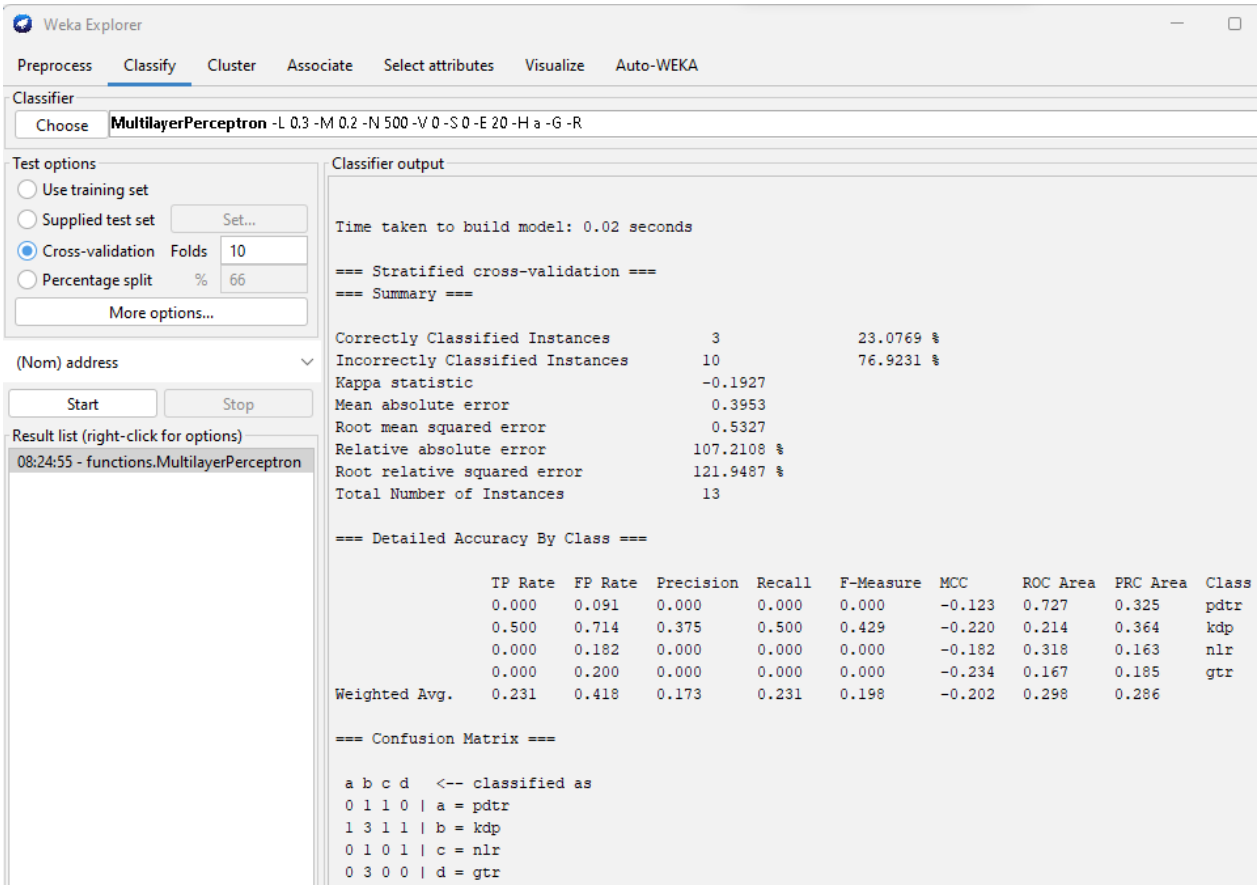
v) Optimize base classifier's performance

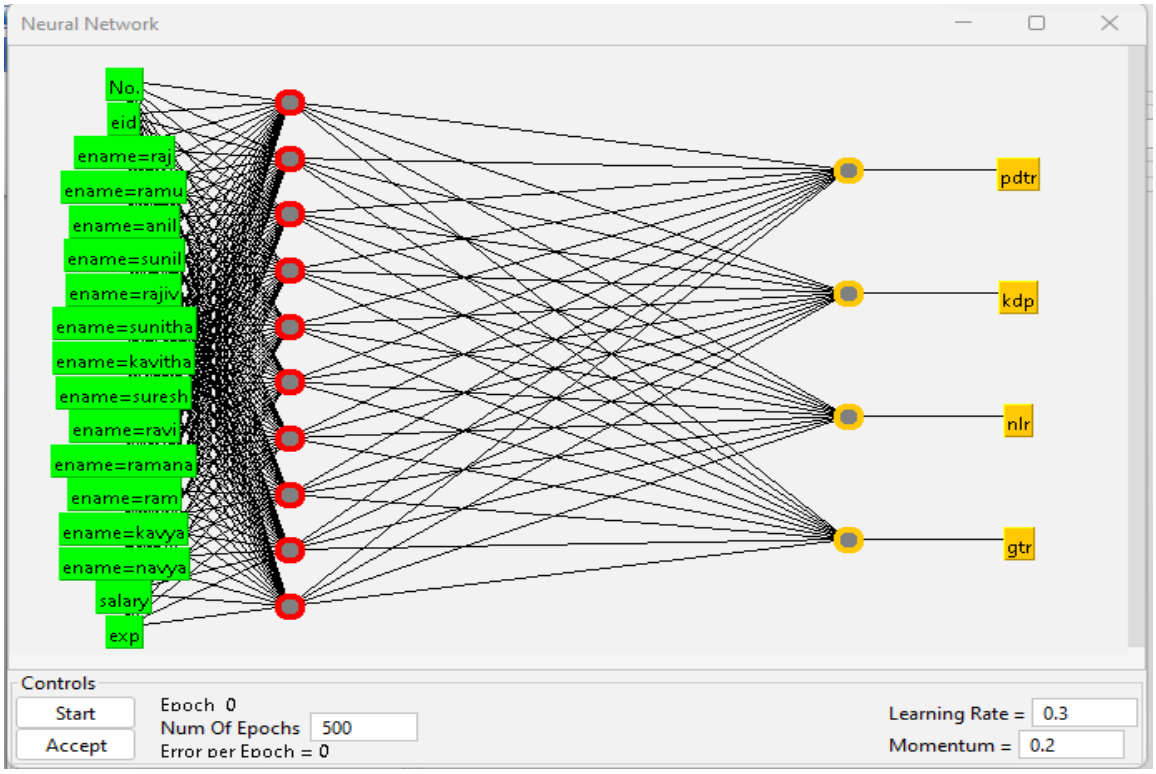
vi) Use clustering algorithm such as Cobweb, and Hierarchical Cluster

vii)Select attribute by specifying an evaluator and a search method

i) Classify , Invoke MultiLayerPerception

- **Go to the "Classify" Tab**: This is where you can apply different classification algorithms.
- **Select MultiLayerPerceptron**: Under the "Classifier" section, click on the "Choose" button, then navigate to functions → MultilayerPerceptron.
- **Configure Parameters**: Click on the MultilayerPerceptron name to open the configuration panel. Here, you can set parameters like the number of hidden layers and epochs.
- **Run the Classifier**: Once the settings are configured, click "Start" to train the neural network with the MultilayerPerceptron.



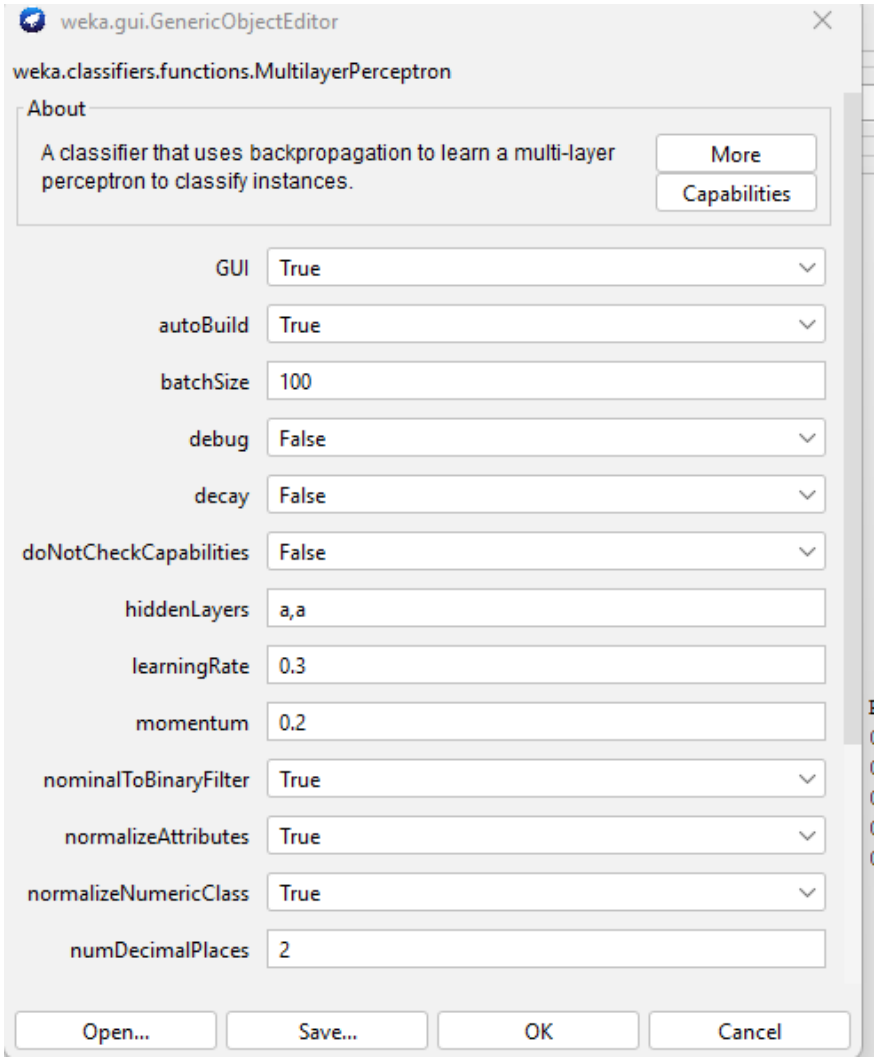


ii) Build neural network GUI as below

a) Beginning the process of editing the network to add a second hidden layer

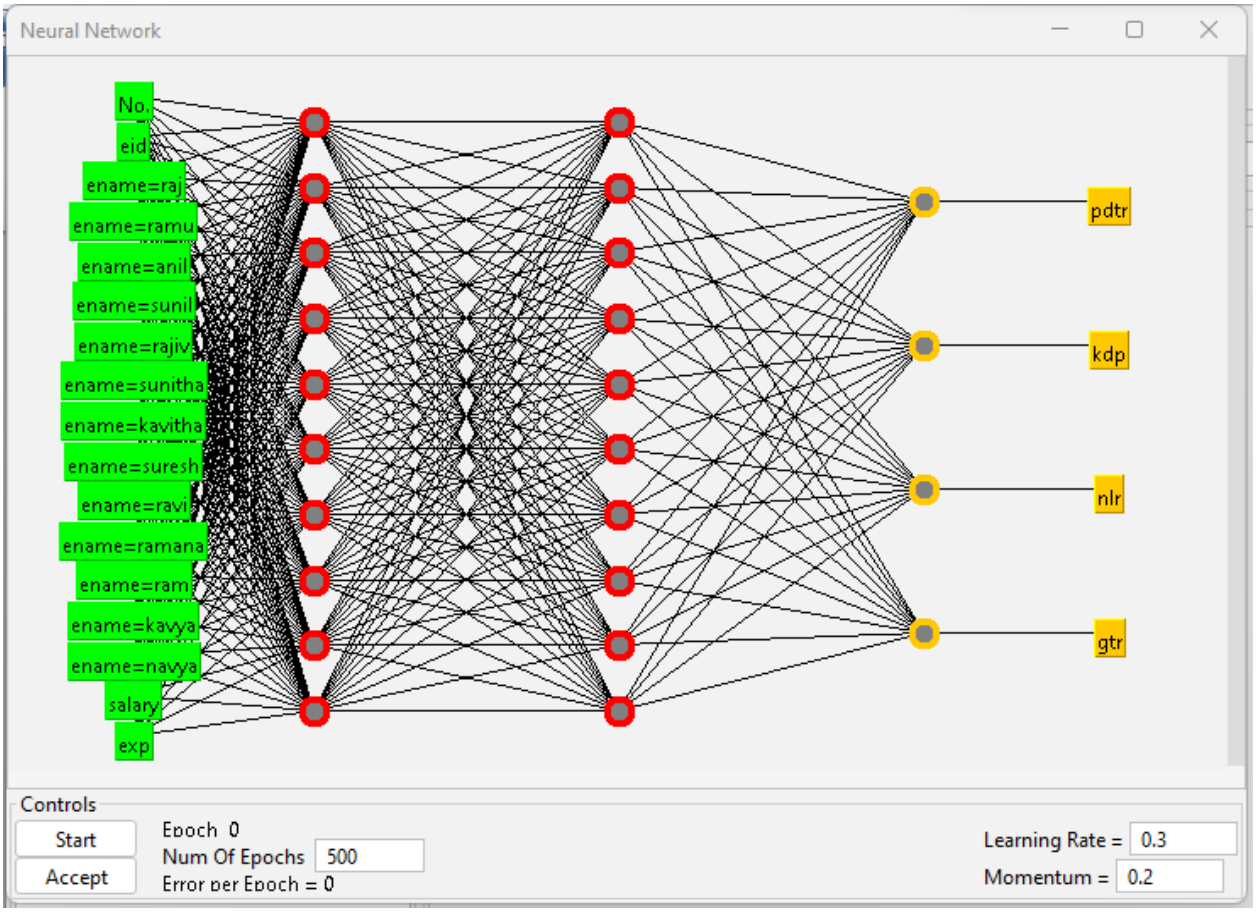
b) The finished network with two hidden layers

- **Edit the Network**: Modify the hiddenLayers parameter in the MultiLayerPerceptron setup to create a network with two hidden layers. Each "a" represents a hidden layer with adaptive size.
- **To add a second hidden layer**, set the hiddenLayers parameter to include two layers (e.g., "a,a" to specify two hidden layers with adaptive sizes).
- **Check the Final Network Structure**: After running the classifier, you can view the results and structure details under the "Result list" section.
- **Run the Classifier**: Once the settings are configured, click "Start" to train the neural network with the MultilayerPerceptron.



Classifier output

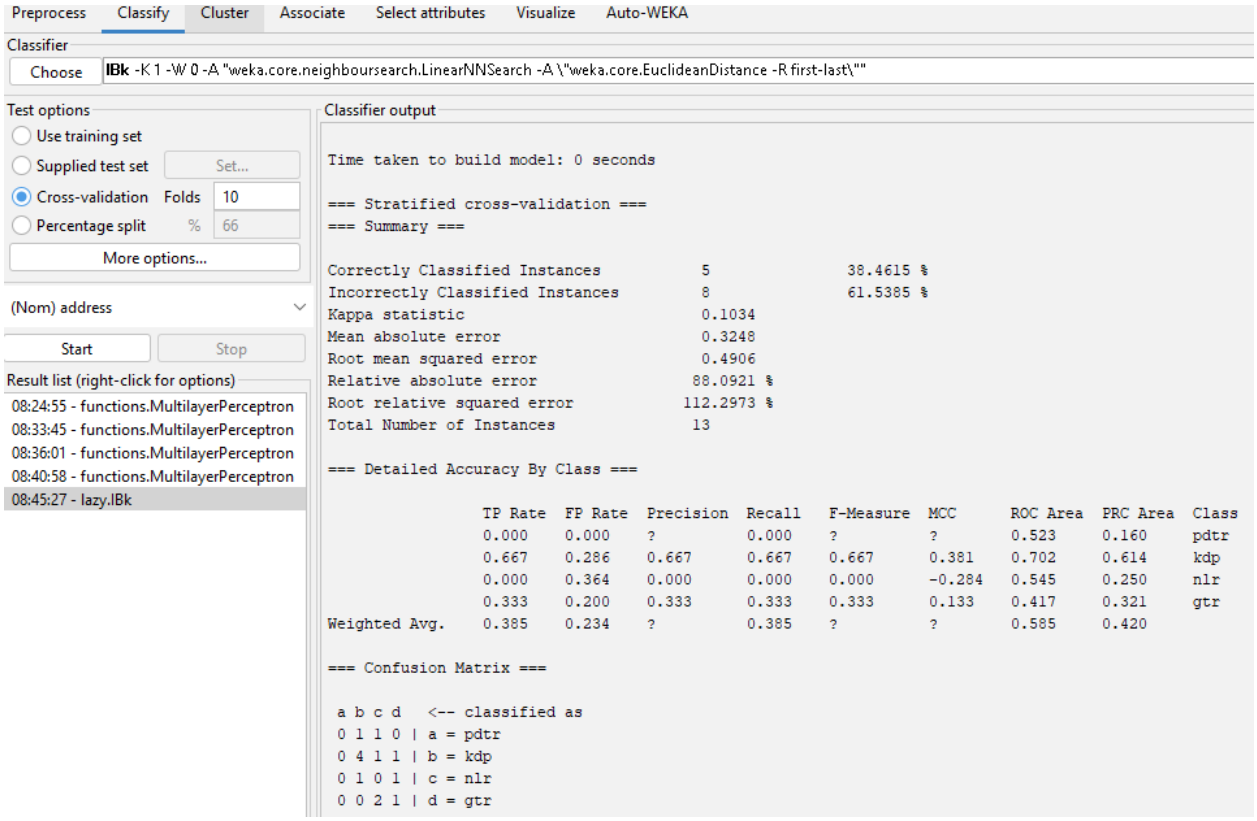
```
=== Run information ===

Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H "a, a" -G -R
Relation:     lab 3 db
Instances:    13
Attributes:   6
              No.
              eid
              ename
              salary
              exp
              address
Test mode:    10-fold cross-validation
```



iii) Apply Lazy classifier, multi instance classifier

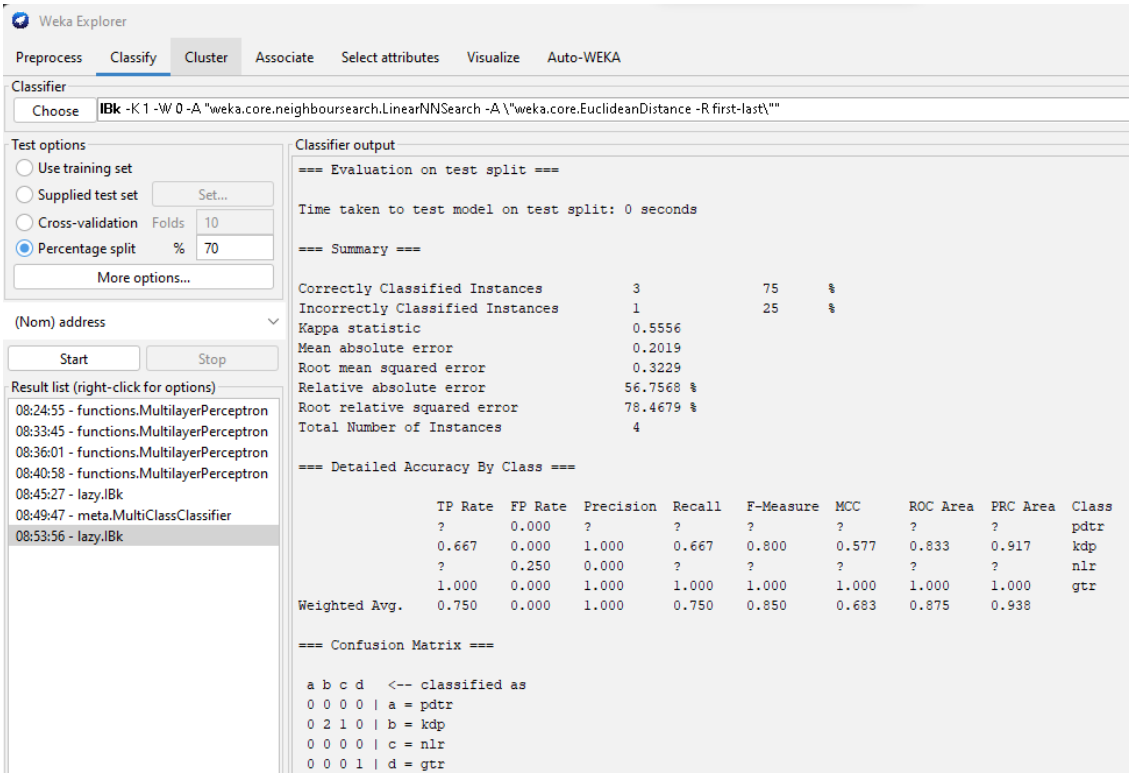
Lazy Classifier:

- **Choose a Lazy Classifier**: Click "Choose" under "Classify" and select lazy → IBk (k-nearest neighbors) as a lazy classifier.
- **Configure and Run**: Set parameters like the value of k and run the classifier by clicking "Start".



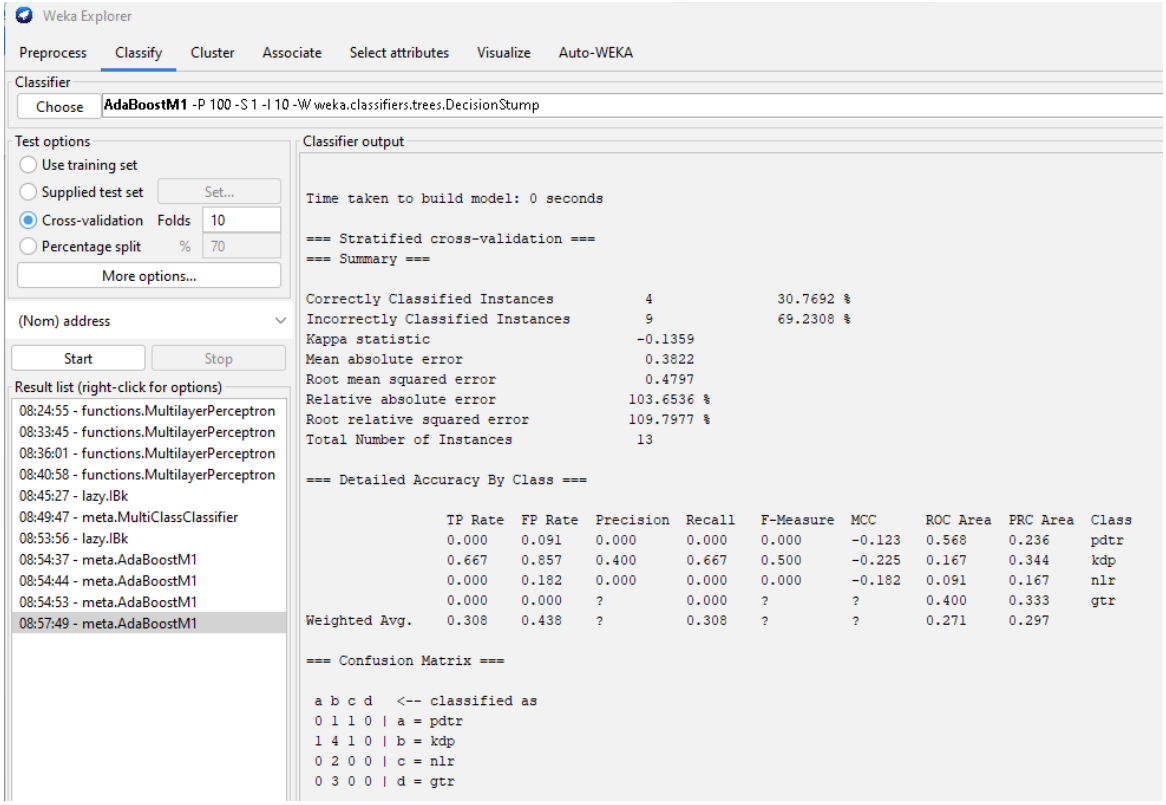
Multi-Instance Classifier:

- **Choose a Multi-Instance Classifier**: Go to meta → MultiClassClassifier.
- **Set Parameters**: You can specify a base classifier within this setup.
- **Run**: Click "Start" to execute the multi-instance classifier.



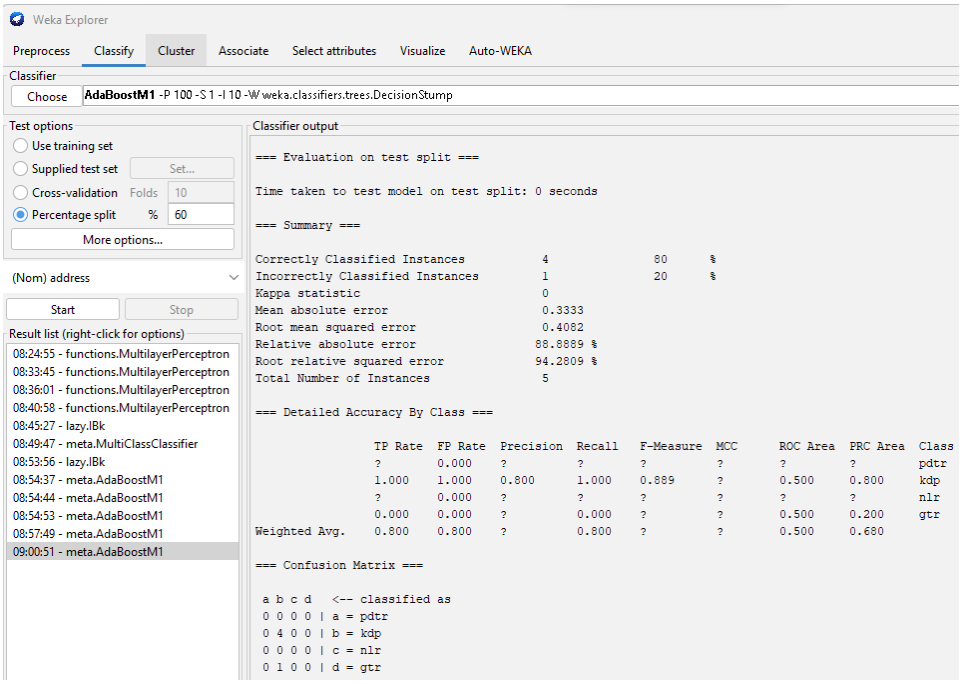
iv) Apply any MetaLearning Algorithm

- **Select a Meta-Classifier**: Meta-learning algorithms in Weka can be found under `meta` in the "Choose" menu. Examples include `Bagging`, `Boosting`, `Stacking`, and `AdaBoostM1`.
- **Configure and Run**: Choose one of the meta-learning algorithms, configure its parameters as needed, and click "Start".



v) Optimize base classifier's performance

- **Parameter Tuning**: Many classifiers allow for parameter tuning to improve their performance. You can adjust settings in the classifier configuration window.
- **Use Cross-Validation**: In the "Test options" section, select k-fold cross-validation (usually 10-fold) to ensure better performance measurement.



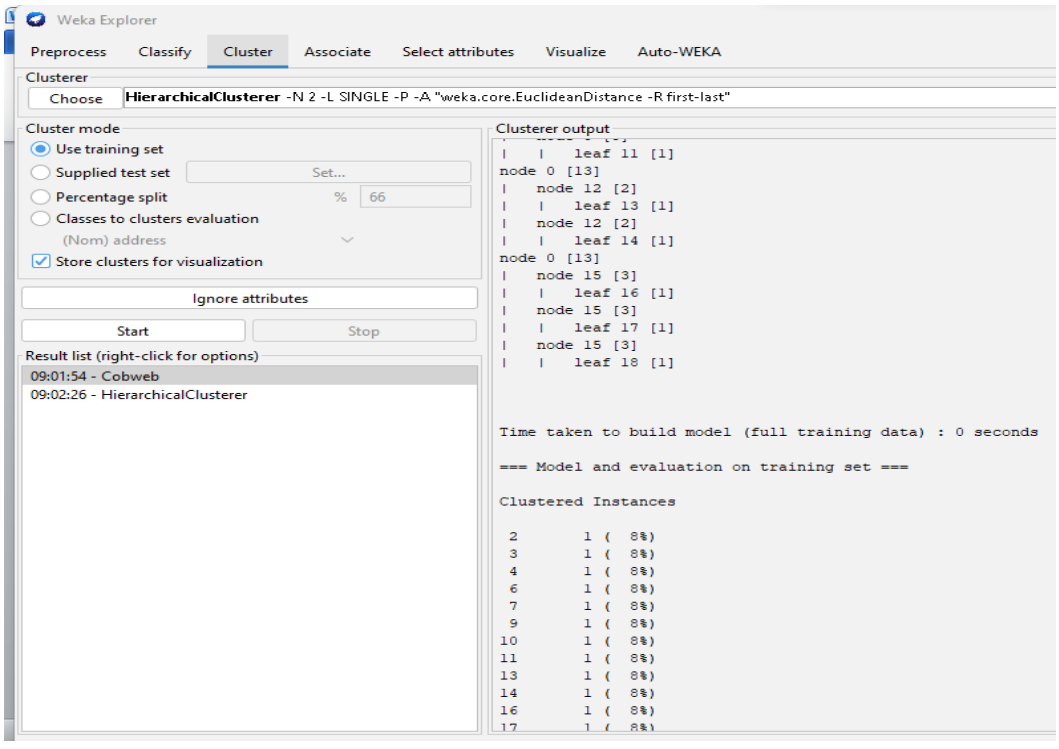
vi) Use clustering algorithm such as Cobweb, and Hierarchical Cluster

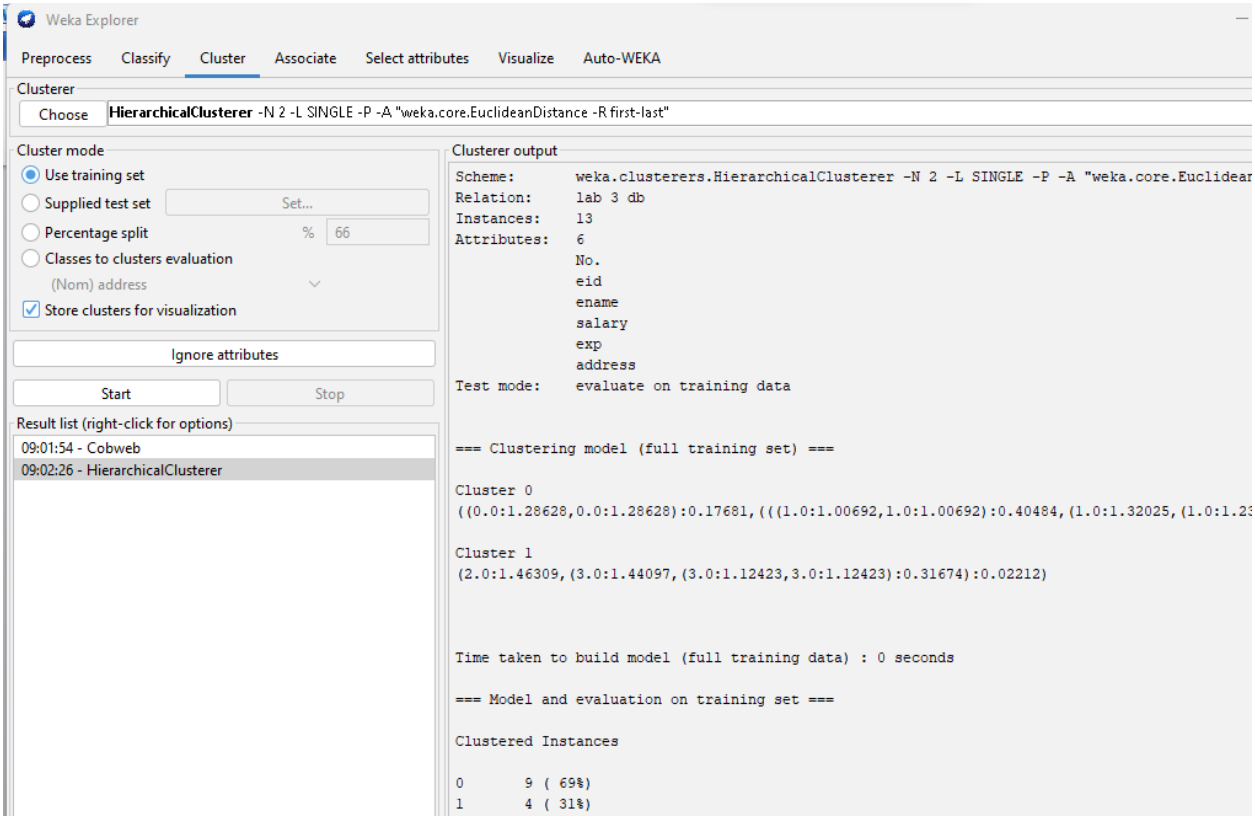
- **Go to the "Cluster" Tab**: Switch from "Classify" to "Cluster".
- **Cobweb**:

  Select Cobweb from the list of clustering algorithms.Set parameters as needed, then click "Start" to cluster the dataset.

- **Hierarchical Cluster**:

  Choose HierarchicalClusterer.Configure it (e.g., setting the distance function) and run the clustering.





vii) Select attribute by specifying an evaluator and a search method

- **Go to the "Select attributes" Tab**: In Weka's Explorer, click on "Select attributes".
- **Choose an Evaluator**: Under "Attribute Evaluator", choose an evaluator like CfsSubsetEval or InfoGainAttributeEval to measure attribute importance.
- **Choose a Search Method**: Select a search method like BestFirst or GreedyStepwise.
- **Apply and View Results**: Click "Start" to run the attribute selection process. The results will show the most important attributes based on the chosen evaluator and search method.