**DATA MINING AND VISUALIZATION LABORATORY**

**2. Experiment using WEKA tool.**

**Consider the following data set.**

****

**i). Use the data sources, like ARFF, XML ARFF files.**

**Prepare Dataset in ARFF Format:**

@relation employee

@attribute eid numeric

@attribute salary numeric

@attribute exp numeric

@attribute address {pdtr,kdp,nlr,gtr} % class attribute

@data

101.0, 15000.0, 4.0, pdtr

102.0, 15000.0, 5.0, kdp

103.0, 12000.0, 3.0, kdp

104.0, 13000.0, 6.0, kdp

105.0, 13000.0, 4.0, kdp

106.0, 14000.0, 6.0, nlr

107.0, 15000.0, 5.0, nlr

108.0, 12000.0, 3.0, gtr

109.0, 12000.0, 3.0, gtr

110.0, 13000.0, 4.0, kdp

111.0, 13000.0, 4.0, kdp

112.0, 14000.0, 5.0, kdp

113.0, 14000.0, 5.0, kdp

**Note:- Save file as employee.arff**

**Load Data in WEKA**

* Open **WEKA Explorer**.
* Click **Open file**, select your employee.arff.

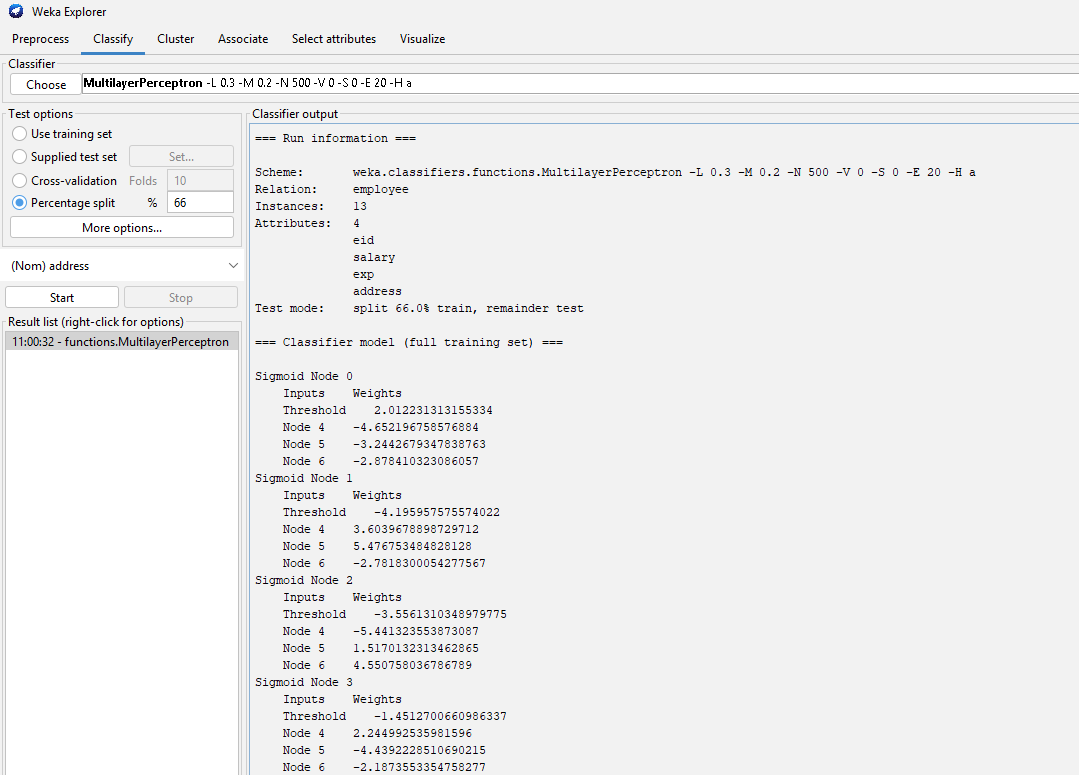
**After Preparing the Dataset Do the following:**

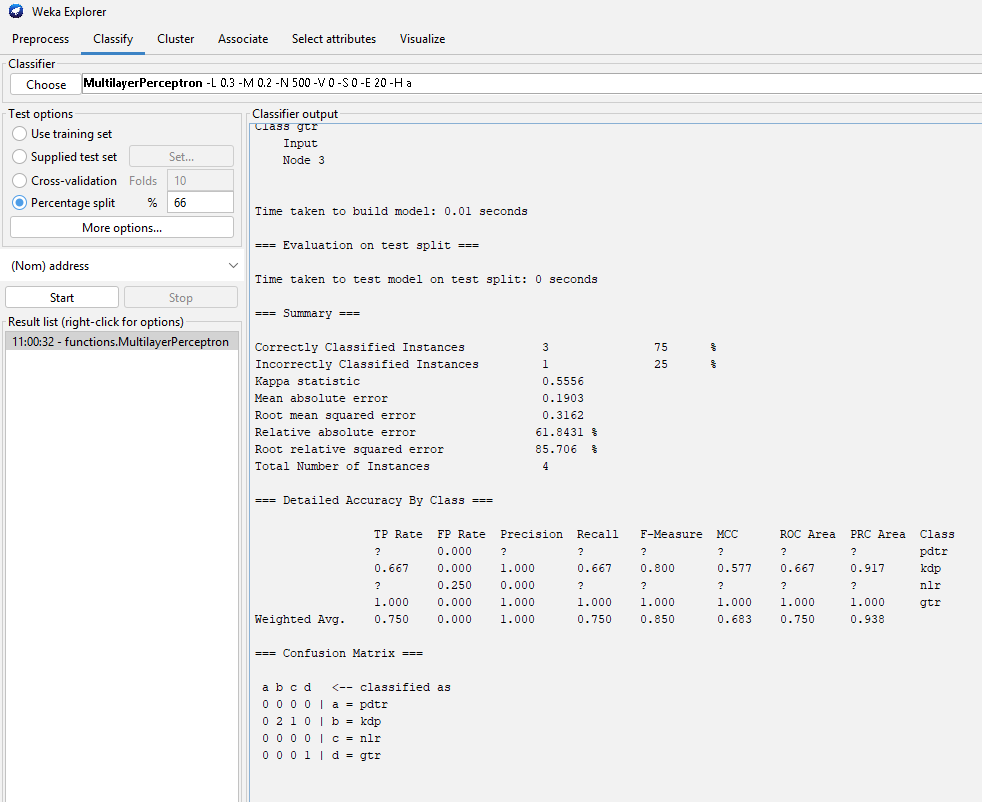
**a). Classify , Invoke MultiLayerPerception.**

**b). Build neural network GUI as below .**

**Build Neural Network**

* Go to the **Classify** tab.
* Choose **Classifier > functions > MultilayerPerceptron**.
* Click **Start** to train with default parameters.
* View output, such as accuracy, confusion matrix.





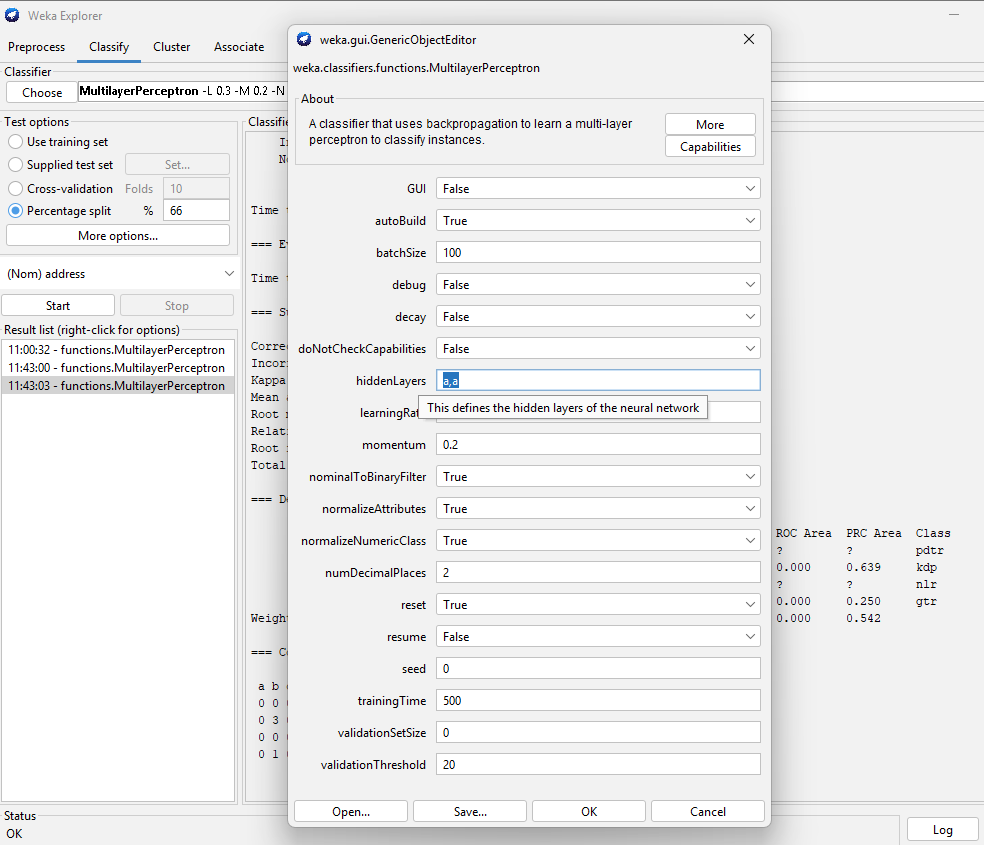
**ii) Beginning the process of editing the network to add a second hidden layer**

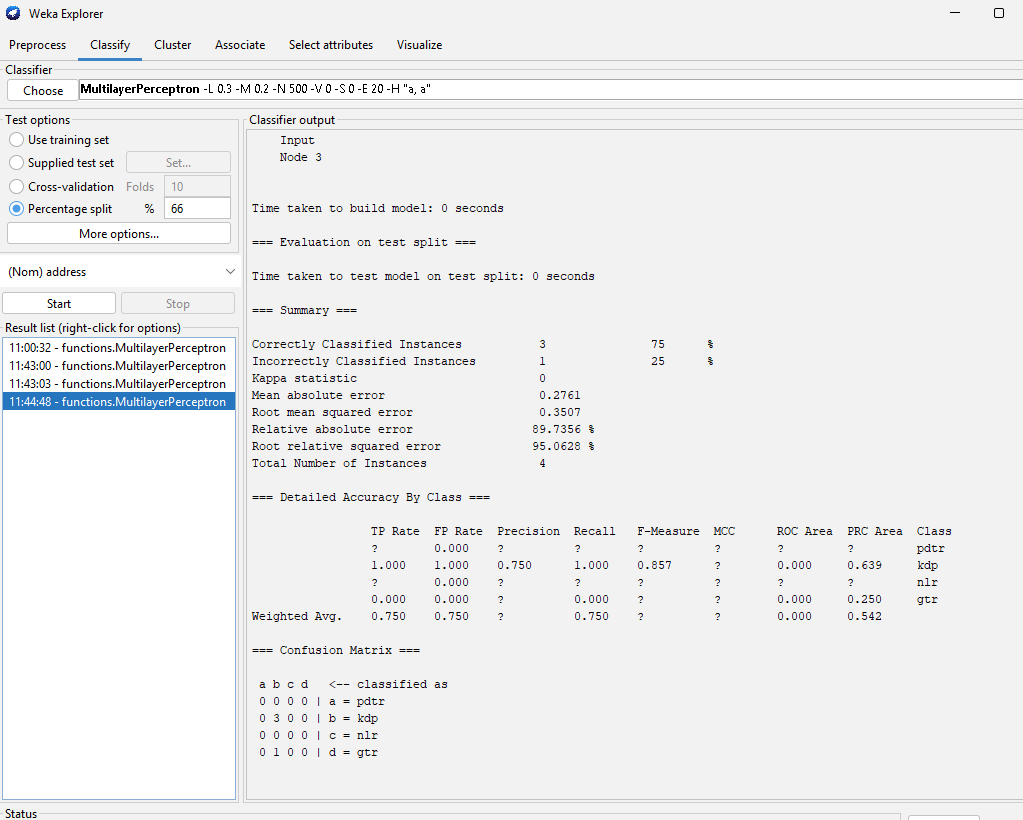
**Edit network architecture:**

* Click on the **MultilayerPerceptron** name → press **Edit**.
* In the “Hidden Layers” field, type a,a (meaning 2 hidden layers, each with number of nodes equal to (attribs + classes) / 2).
* Alternatively, specify exact nodes, e.g., 5,3.
* Click **OK** and **Start**.

**iii) The finished network with two hidden layers.**

* Check output for improved accuracy and structure.





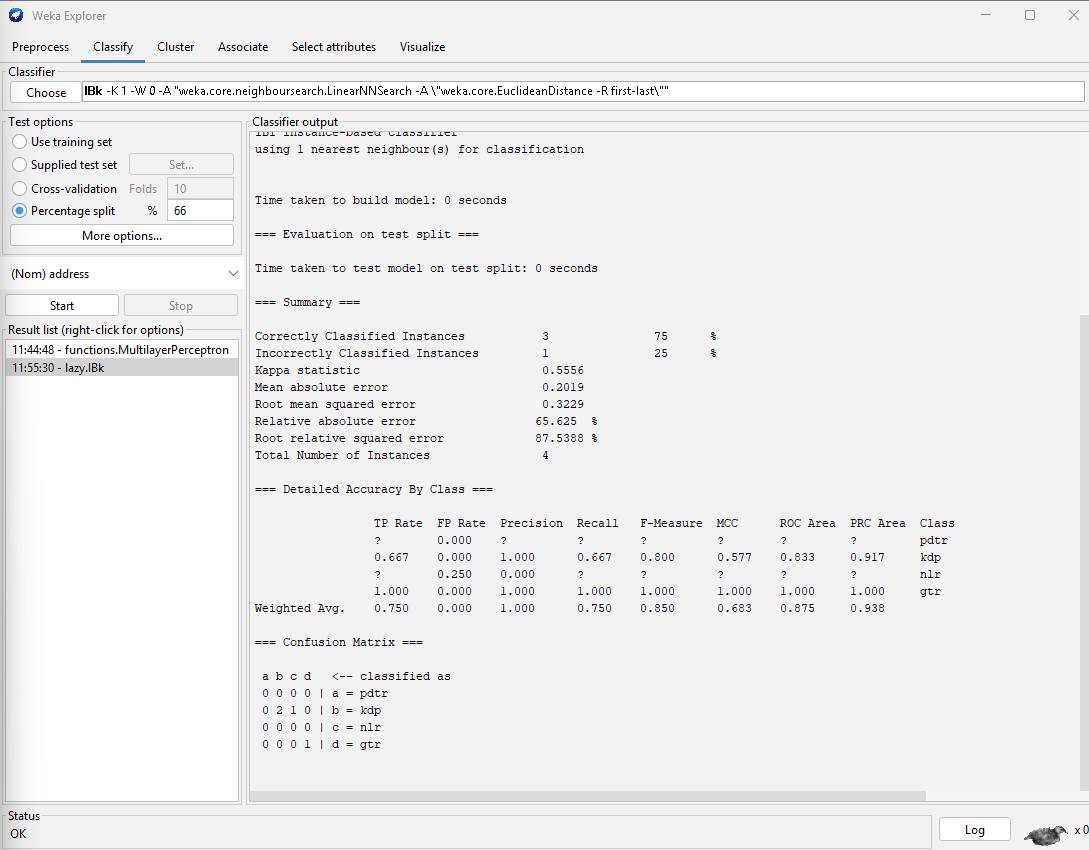
**iv) Apply Lazy classifier, multi instance classifier.**

**Lazy Classifier**

* Go to **Classify** tab.
* Choose **lazy > IBk** (k-NN).
* Click **Start** to train.

**Multi-instance Classifier**

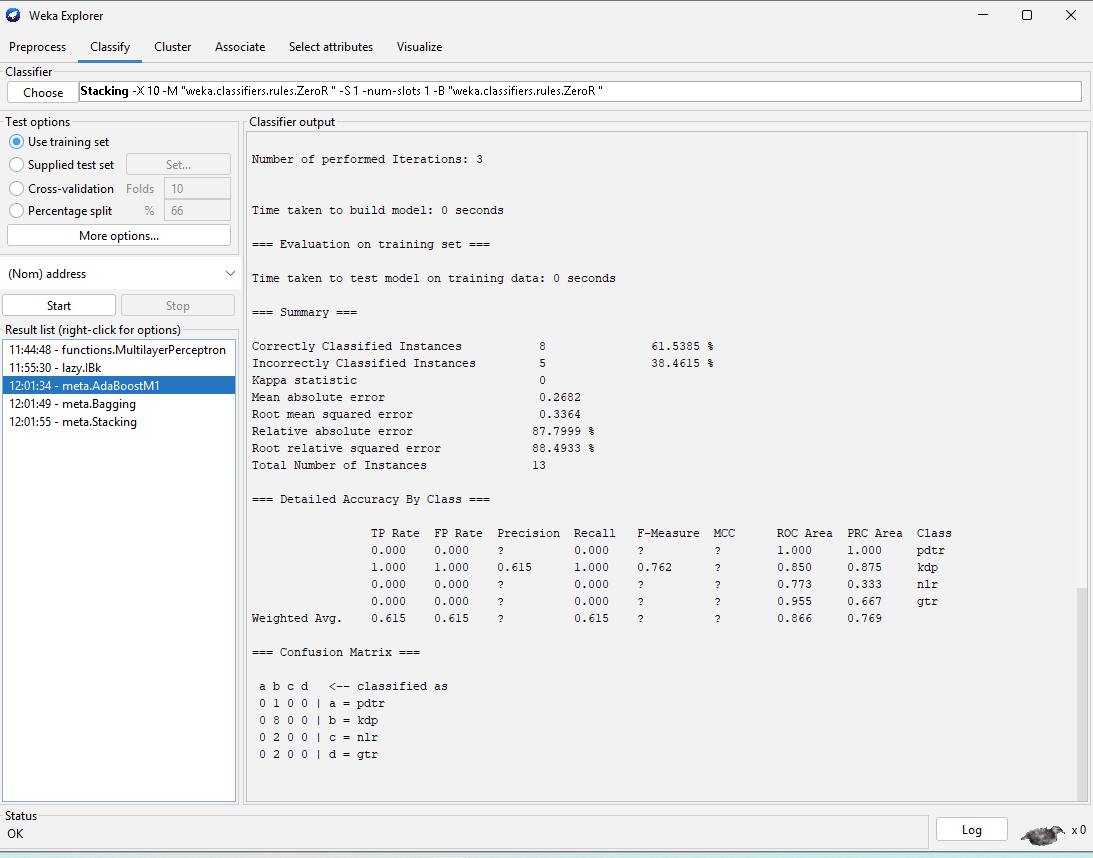
* WEKA’s **multi-instance classifiers** are available under **mi > MultiInstance...**.
* Load dataset suitable for multi-instance (typically requires different format).
* Use **multi-instance** options if applicable.



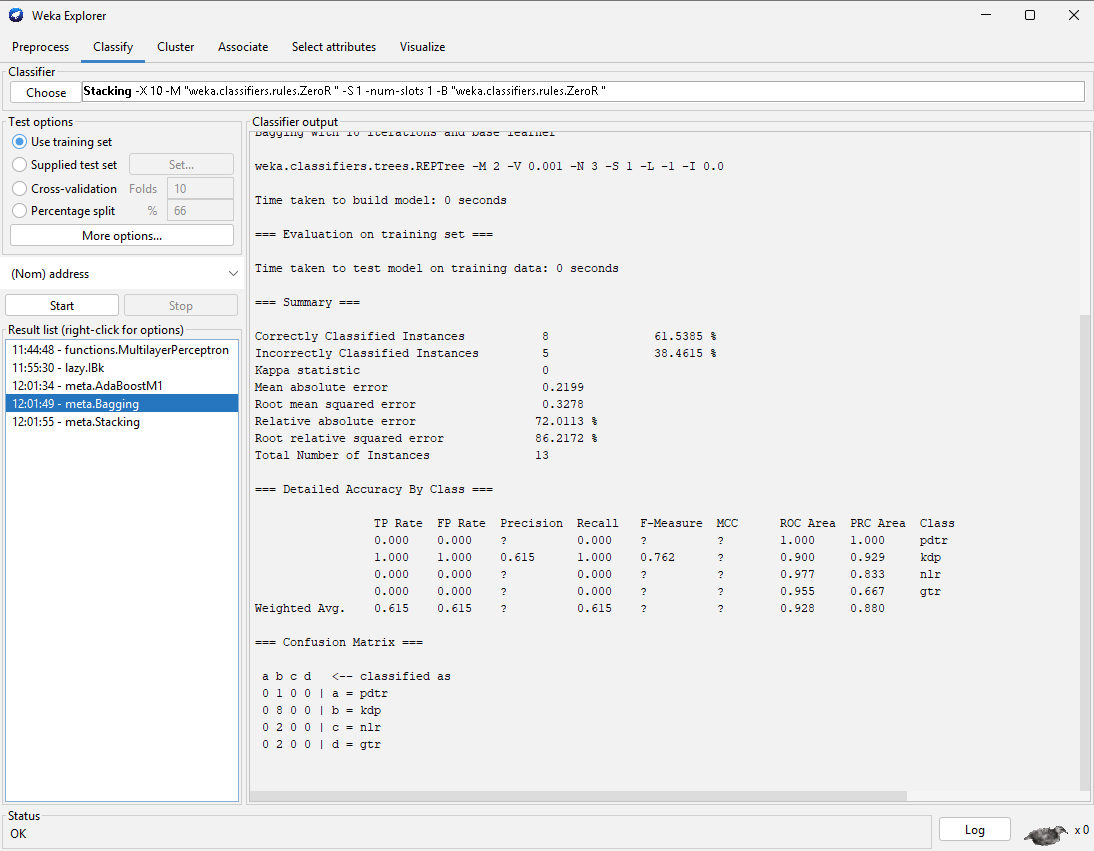
**v) Apply any MetaLearning Algorithm.**

**Meta Classifiers**

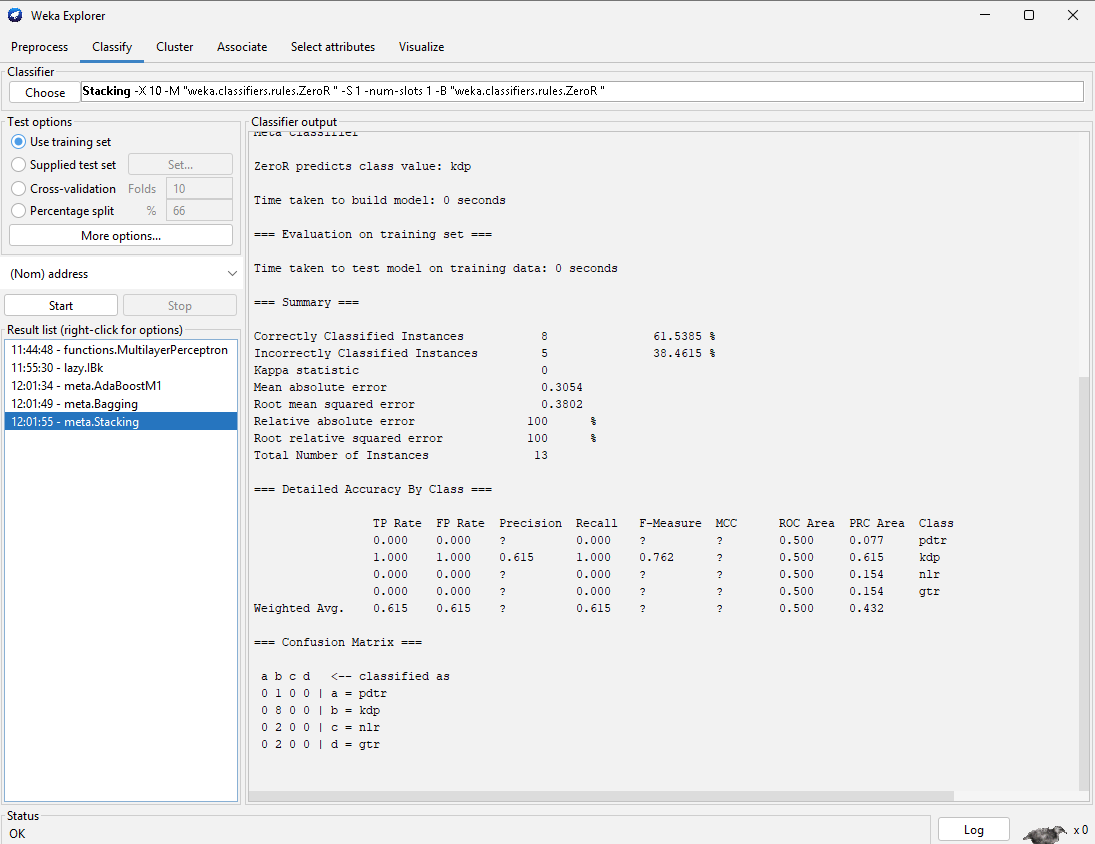
* Under **Classify**, select **meta**.
* Examples: **AdaBoostM1**, **Bagging**, **Stacking**.
* Select one, e.g., **AdaBoostM1**.
* Set base classifier (like J48).
* Click **Start**.
* **AdaBoostM1:**



**Bagging:**

****

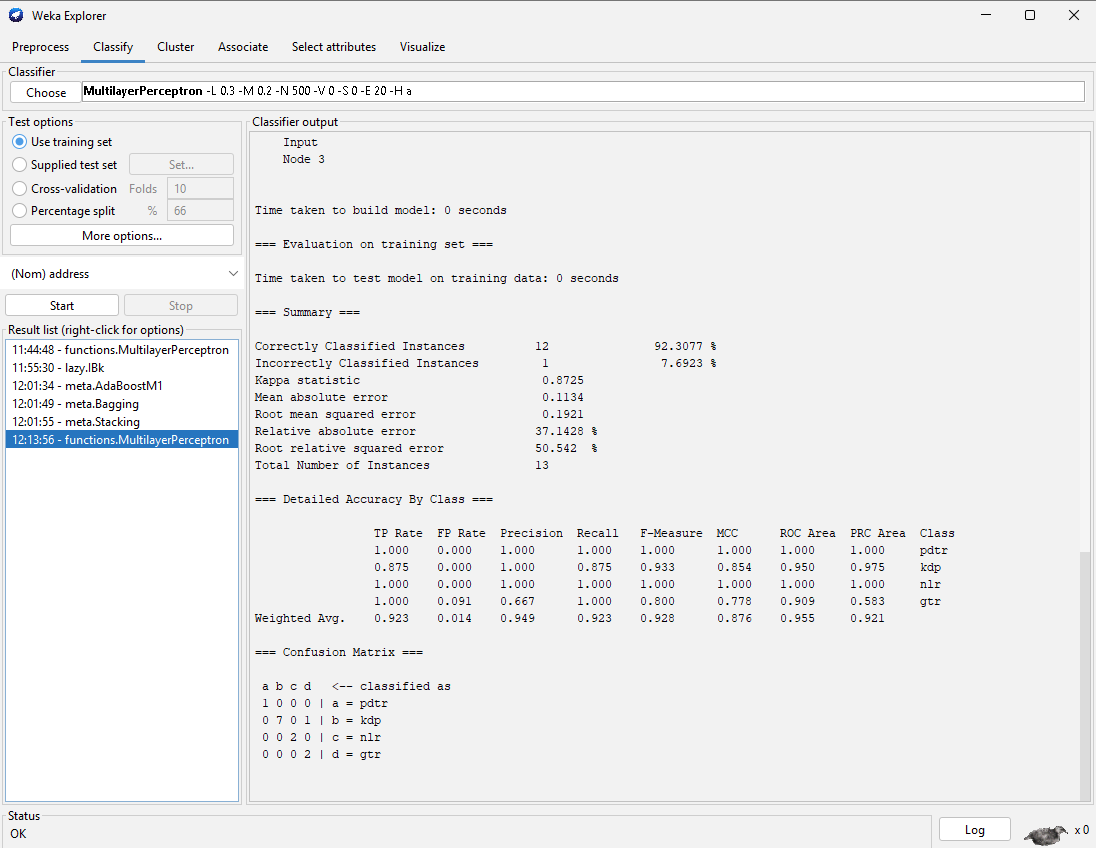
**Stacking:**

****

**vi) Optimize base classifier’s performance.**

Use **Grid Search** or **CV Parameter Selection**:

* Go to **Tools > CVParameterSelection**.
* Choose base classifier (e.g., MultilayerPerceptron).
* Select parameters to tune (e.g., learning rate).
* Run to find optimal parameters.

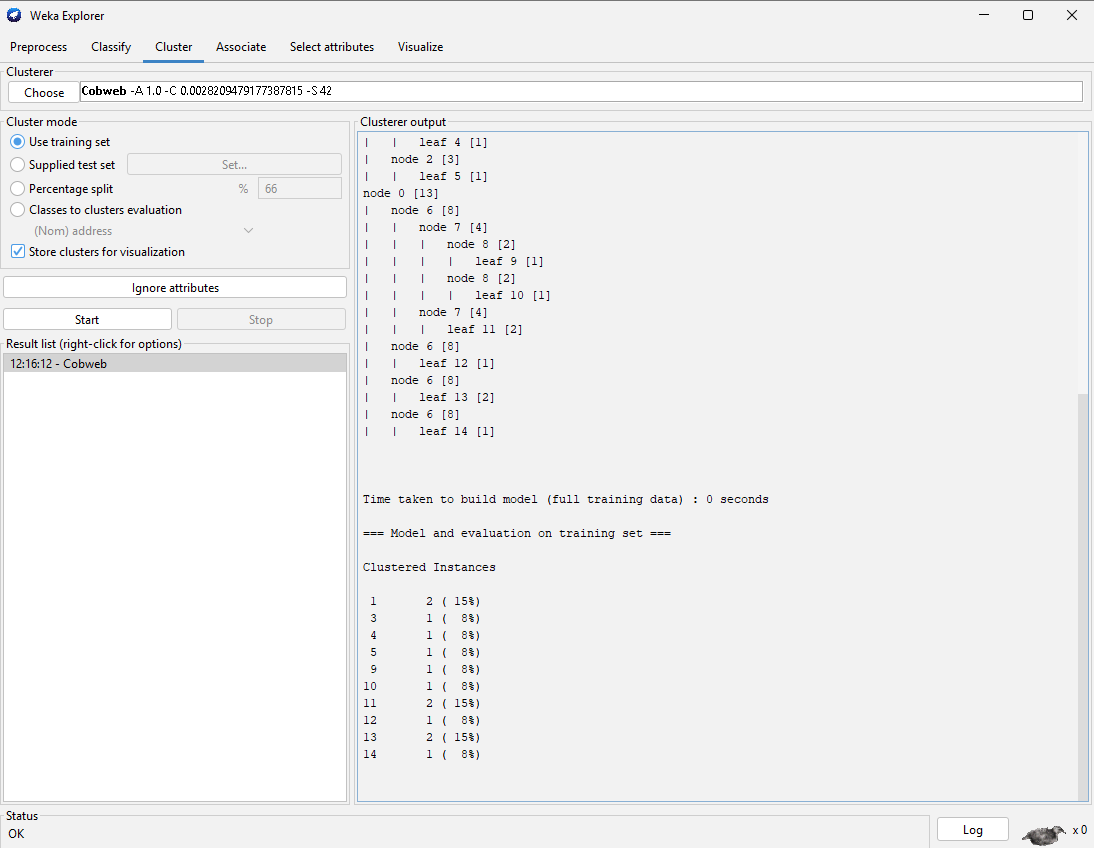


**vii) Use clustering algorithm such as Cobweb, and Hierarchical Cluster.**

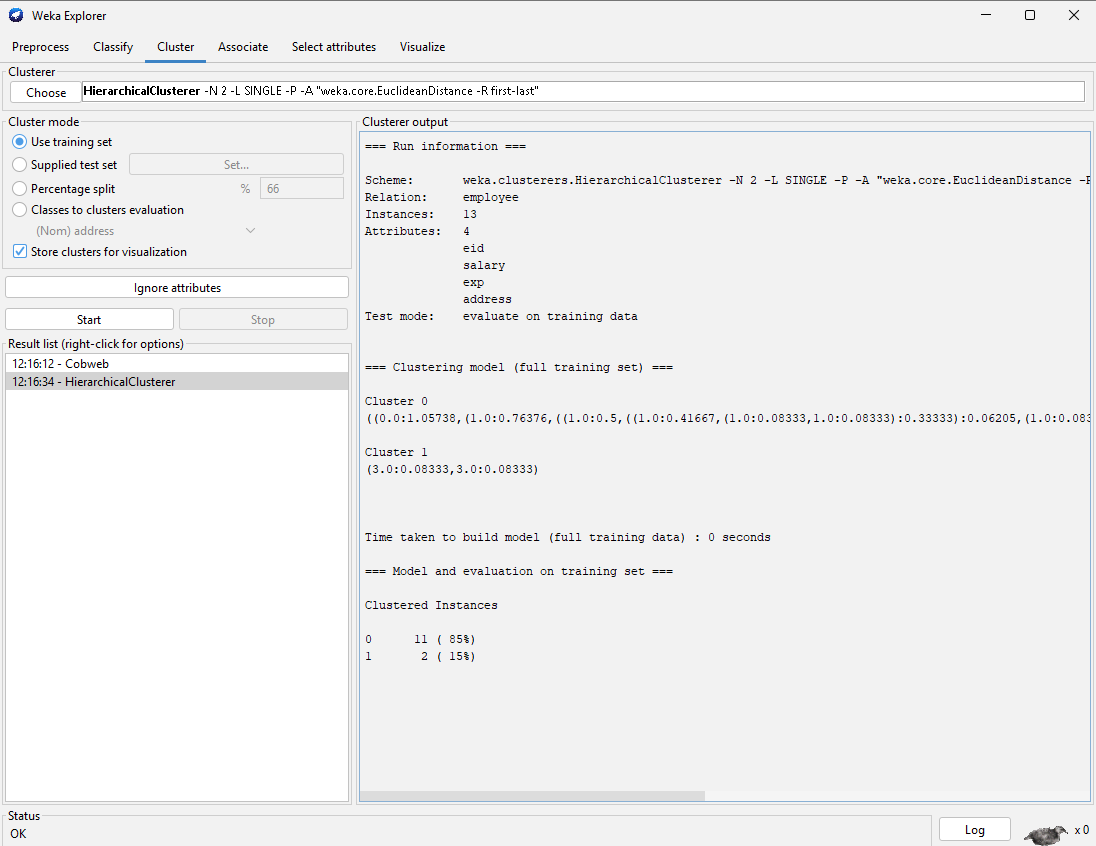
Clustering:

* Go to **Cluster** tab.
* Select **Cobweb**.
* Click **Start** to cluster dataset.
* Repeat with **HierarchicalClusterer**.
* View clusters formed.

**Cobweb:**



**HierarchicalClusterer:**



**viii) Select attribute by specifying an evaluator and a search method.**

**Select Attributes**

* Go to **Select attributes** tab.
* Choose evaluator: e.g., **CfsSubsetEval**.
* Choose search method: e.g., **BestFirst**.
* Click **Start** to select important attributes.

