Neural Networks to predict response

DENG, Yang

ydeng@se.cuhk.edu.hk

Overview

- What is Neural Network?
- Building Neural Network with Weka
- Momentum in Neural Network
- Building Neural Network using GUI
- Clustering Algorithms with Weka

What is Neural Network?

Neural Network is a machine learning model inspired by the biological neural networks that simulate what our brain does.

Neural Network has several components including the Input Layer, Hidden Layers and Output Layer:

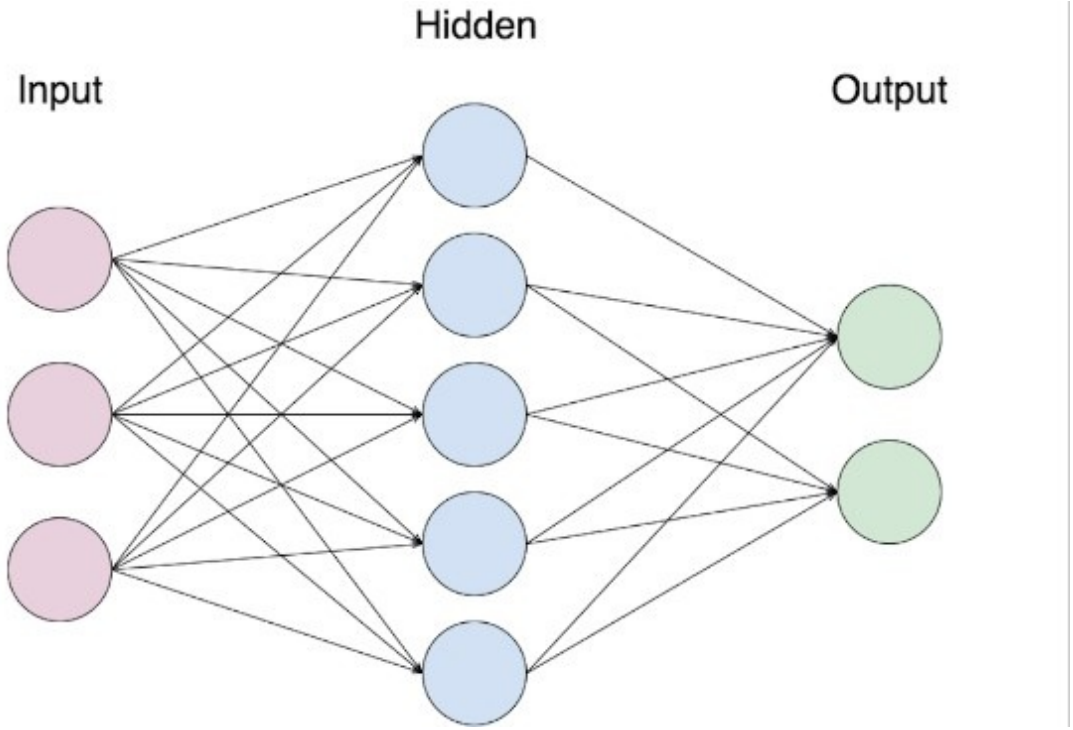
(1) **Input Layer** denotes the input variables that will be fed into the network,

(2) **Hidden Layers** are the computation layers (or parameters) that will be trained,

(3) **Output Layer** denotes the output of the model. For example, the class label in classification task or the real number in regression task.

What is Neural Network?

A typical neural network model can be represented as follow:



What is Neural Network?

As you can see in the figure, each layer contains different number of nodes.

Usually, the number of nodes in input layer will be the number of how many input variables while the number of nodes in output layer will be the number of outputs of the model, i.e, number of classes you have in classification.

The number of nodes in each hidden layer is specified by user.

What is Neural Network?

Besides the structure of the neural network, there are another two core parts of the network, **activation function** and **cost function**.

Activation Function is one of the most powerful cores, which is responsible for powering Neural Networks. In other words, it decides which neurons will be activated. Just like our brain, it determines what information would be passed to further layers.

Without activation functions, the neural networks would not have such a meaningful representative power.

What is Neural Network?

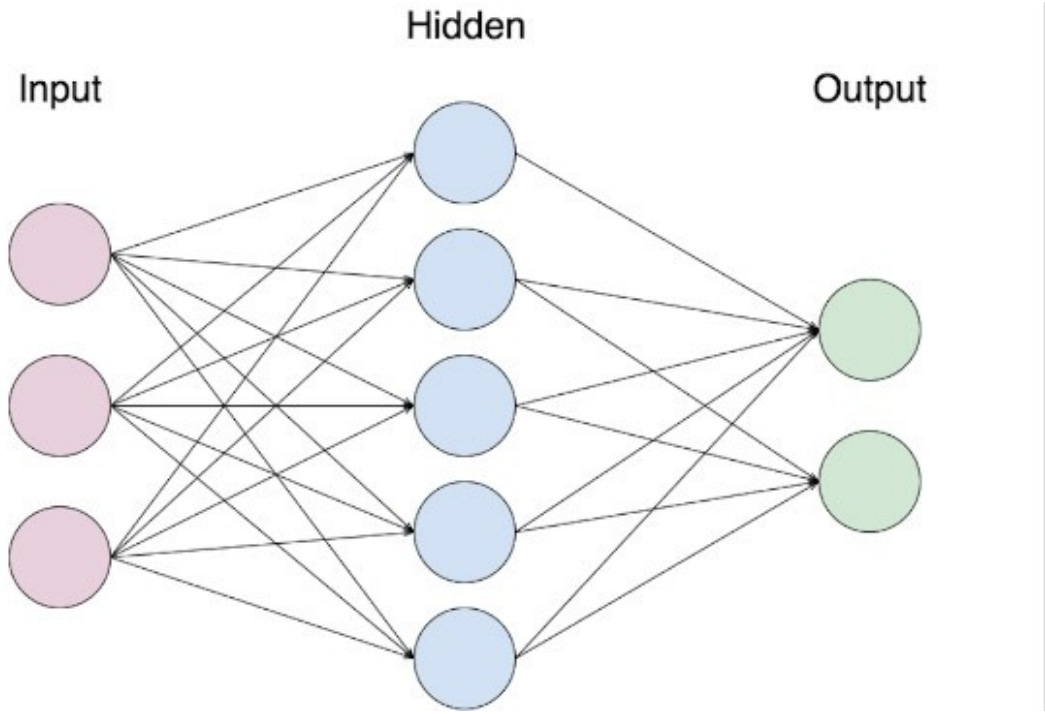
Sigmoid function, Hyperbolic tangent function and ReLu function are the activation functions widely used in modern neural network models.

Another core part is the **cost function**. It is used to calculate loss given the true and predicted results. The aim of the neural network is to minimize this loss. So cost function is to evaluate how well is our neural network and effectively drives the learning of neural network towards its goal.

Some of the most famous cost functions are Cross Entropy, which is used in classification task, and Mean Square Error, which is used in regression task.

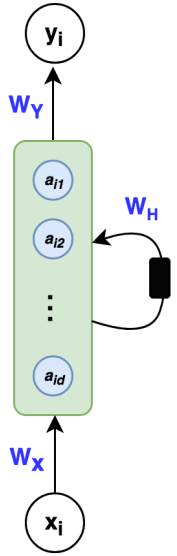
What is Neural Network?

Back to this figure, the network with this structure is called **Feed-Forward network** or **Multi-layer perceptron**.

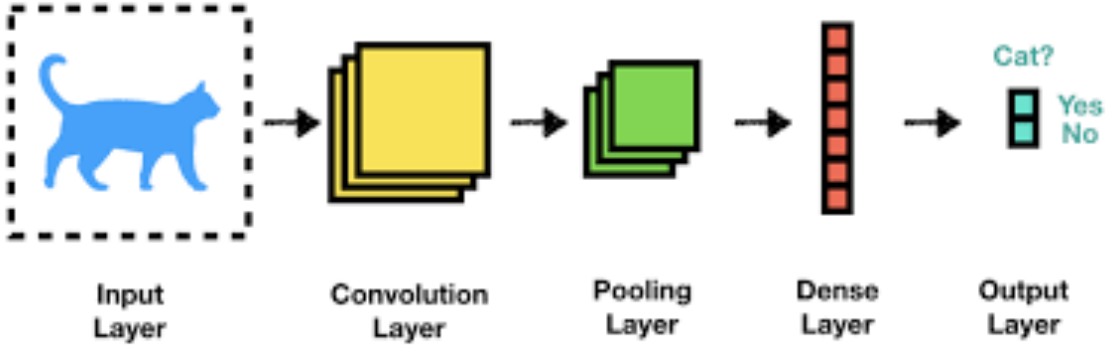


What is Neural Network?

There are some advanced neural networks with special structure. For example, the **recurrent neural network (RNN)** that can be used to handle time-series (weather of each day) or sequential data (text). Or **convolutional neural network (CNN)** that is widely used for handling image.



Recurrent Neural Network



Convolutional Neural Network

What is Neural Network?

Also, there are many transformations of these neural networks. For example, RNN was developed into **Gated Recurrent Unit (GRU)** and **Long-Short Term Memory (LSTM)**. And there are many other advanced neural networks with different structure proposed in recent years, like Ca**psule Network**, **Graph Neural Network (GNN)** and so on.

Neural Network in Weka

In this tutorial, we will focus on building the Multi-layer perceptron using Weka.

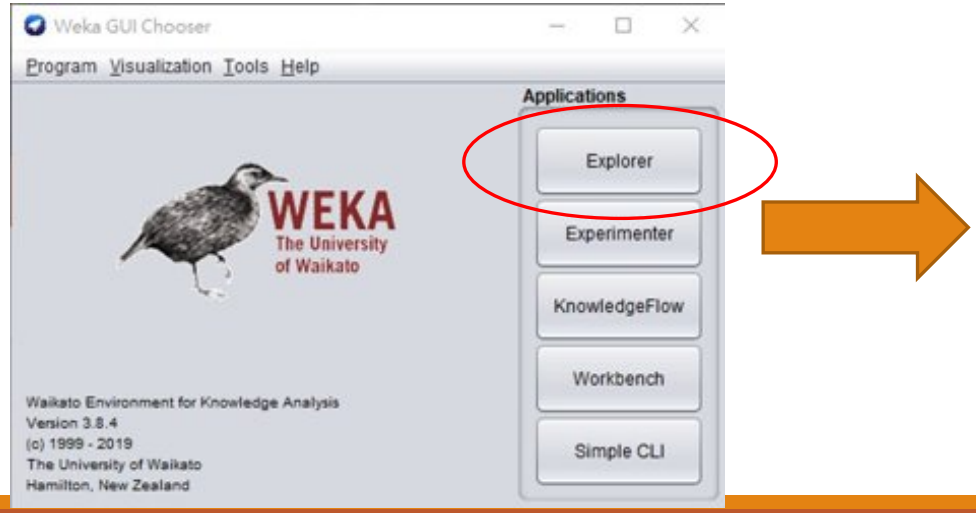
You can use Weka to easily construct a neural network and it will help you to configure most of the setting of it like the activation function and cost function.

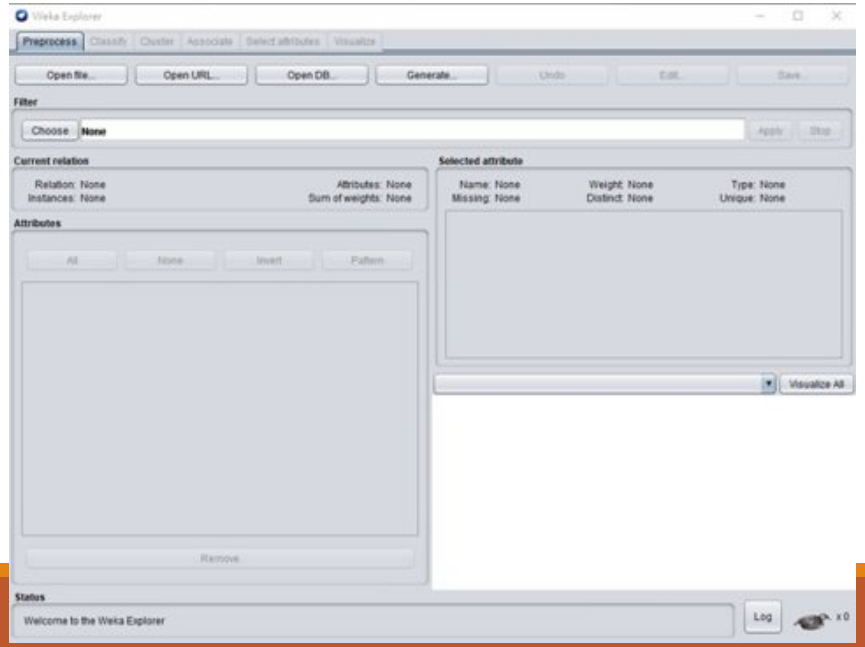
All you need is to prepare the data for it.

Preparation for building Neural Network

Before constructing our neural network, again, we first need to prepare our training data.

Open Weka, choose **Explorer** in the **Weka GUI Chooser**

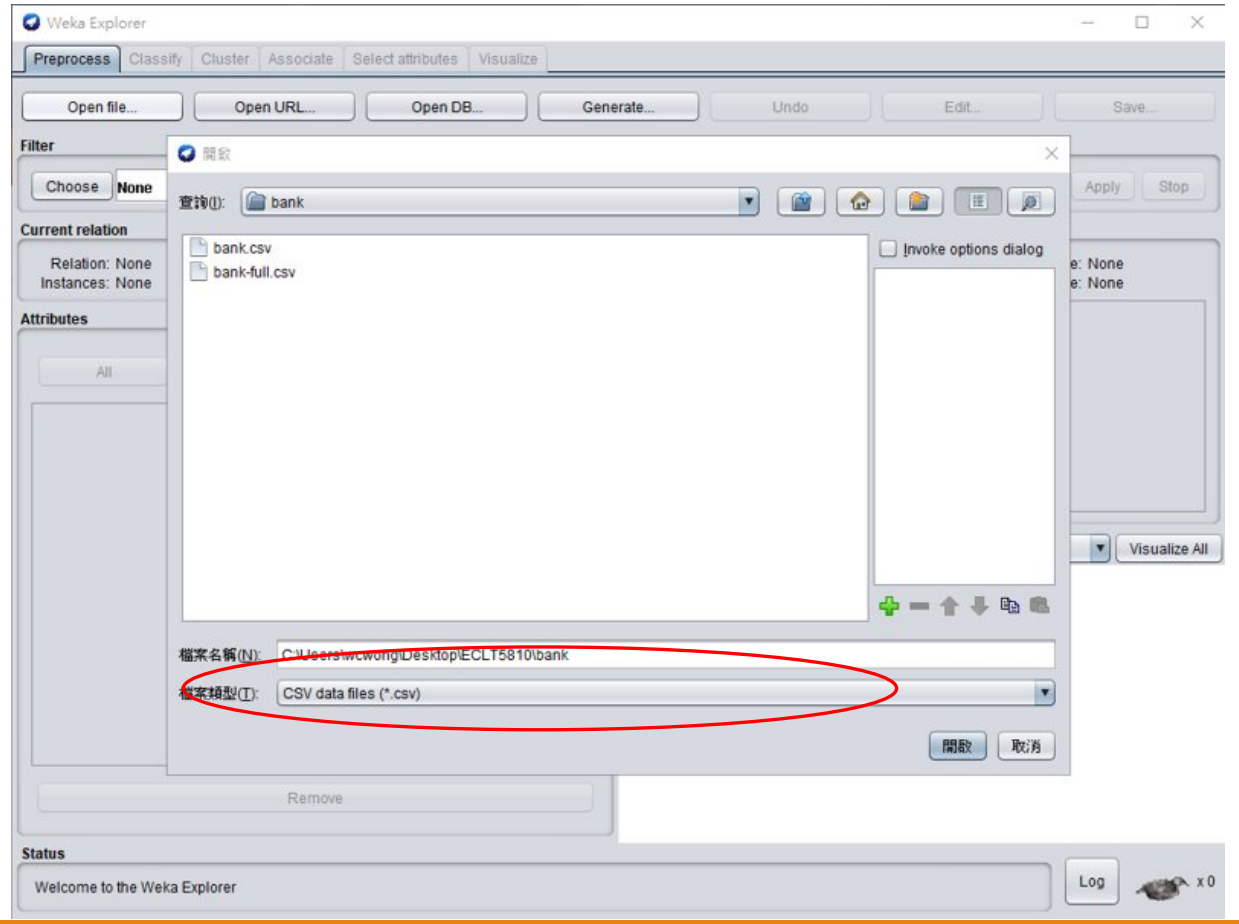




Preparation for building Neural Network

Click **Open file**, then open the bank.csv used in last tutorial

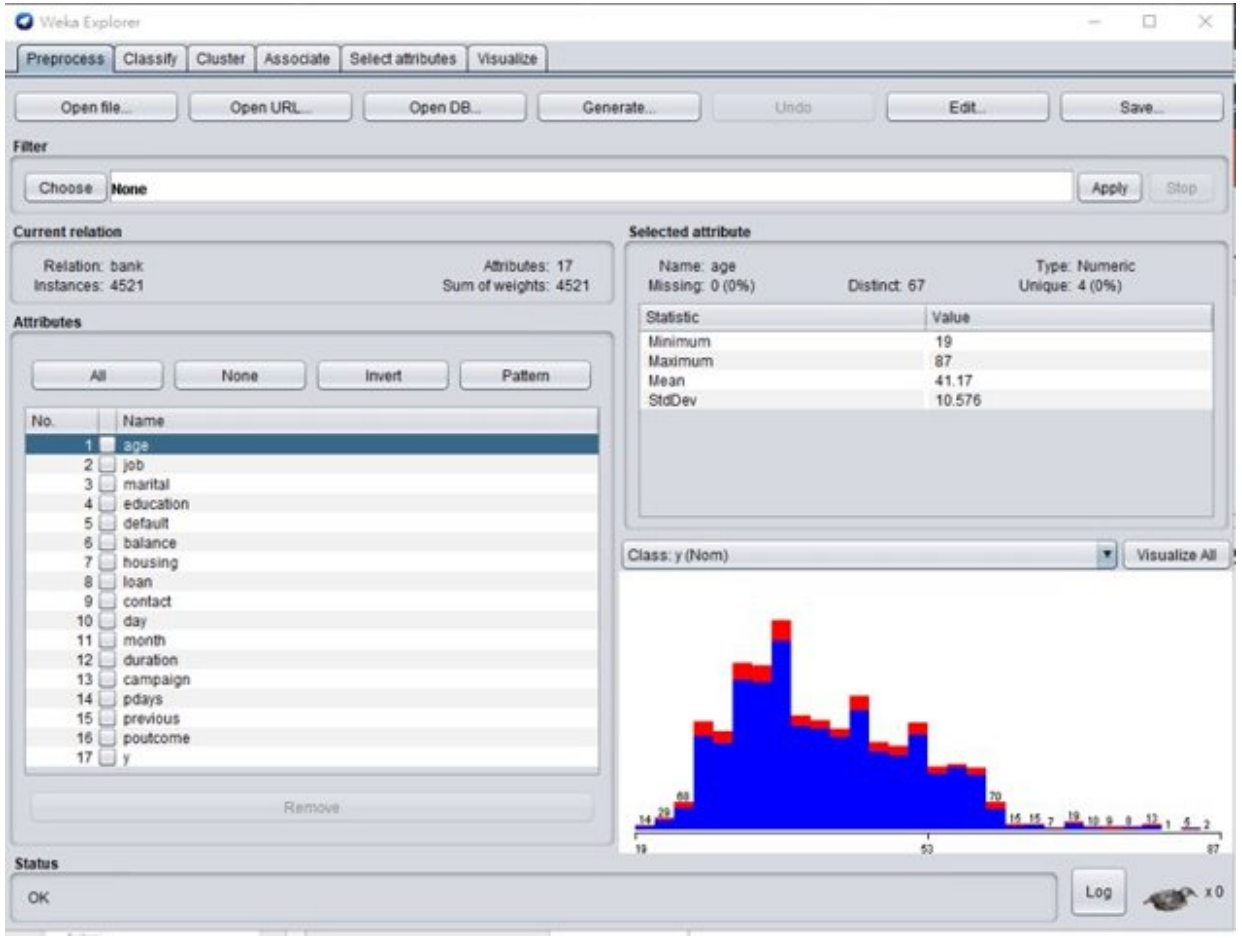
Again, please
remember to change to **CSV data files(*.csv)** in file type.



Preparation for building Neural Network

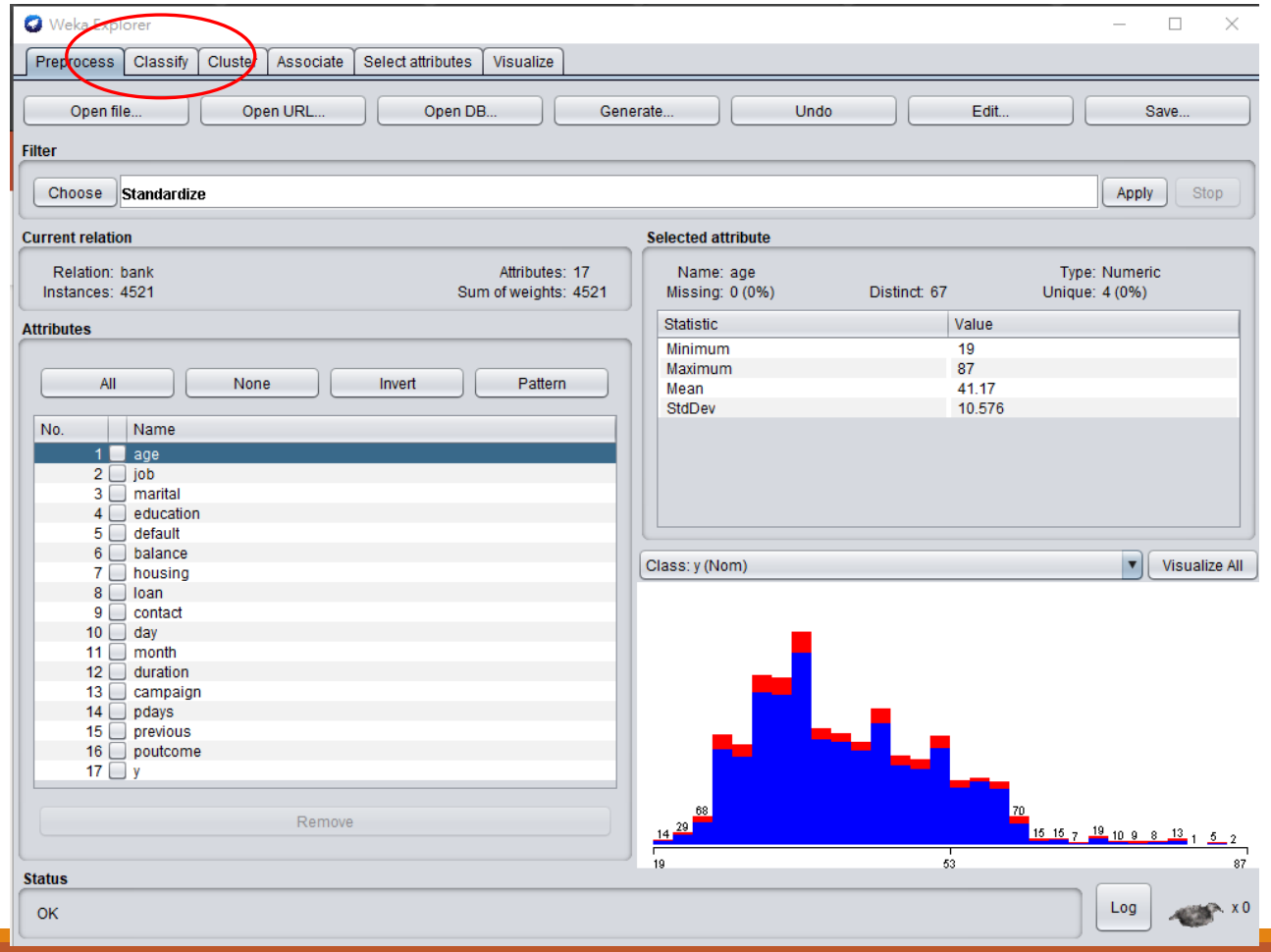
Now, data is loaded into Explorer.

And then we can perform feature engineering before building the Neural Network but this time we simply use the original dataset to do it.



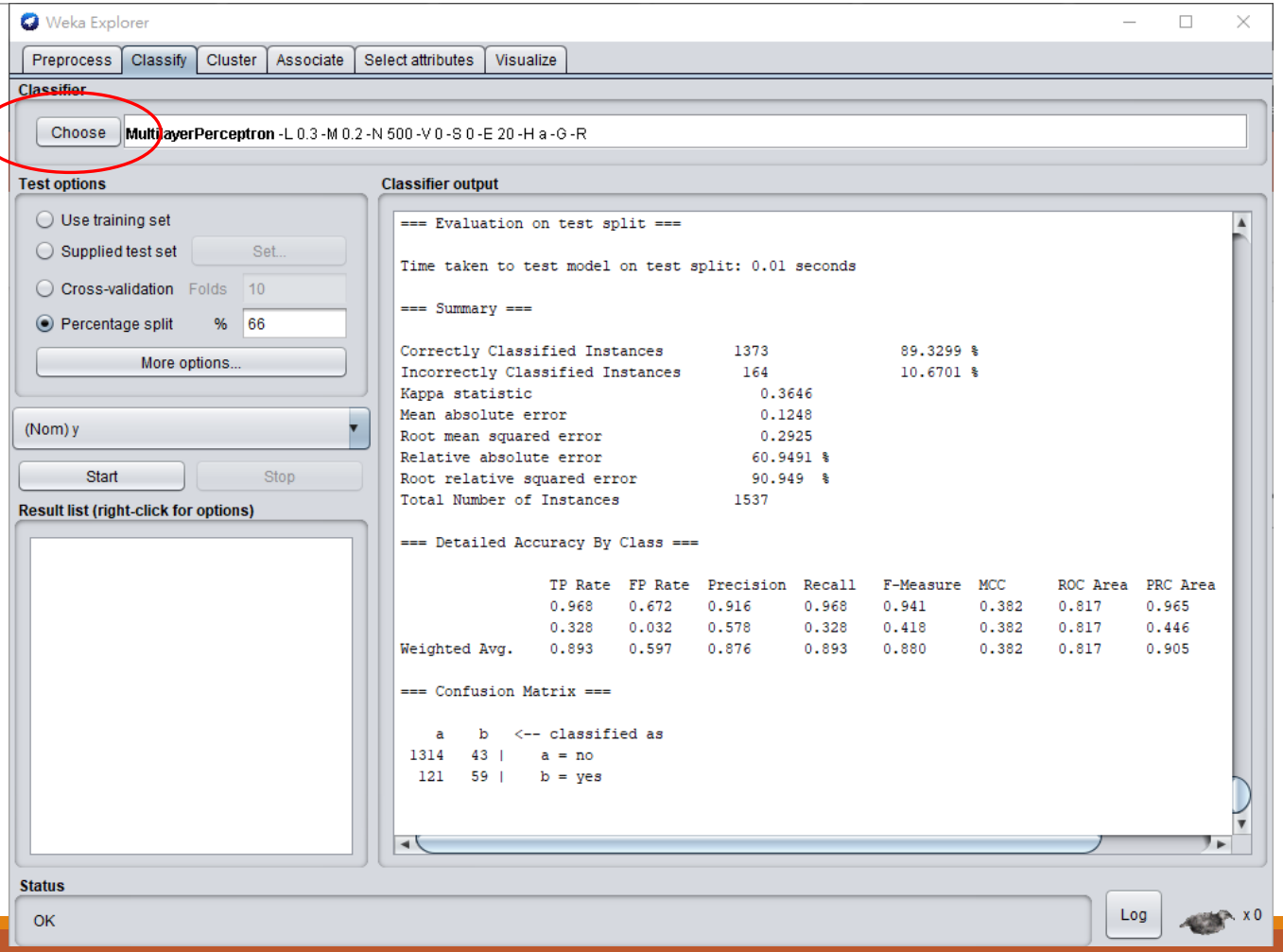
Building Neural Network

Click Classify



Building Neural Network

Click Choose

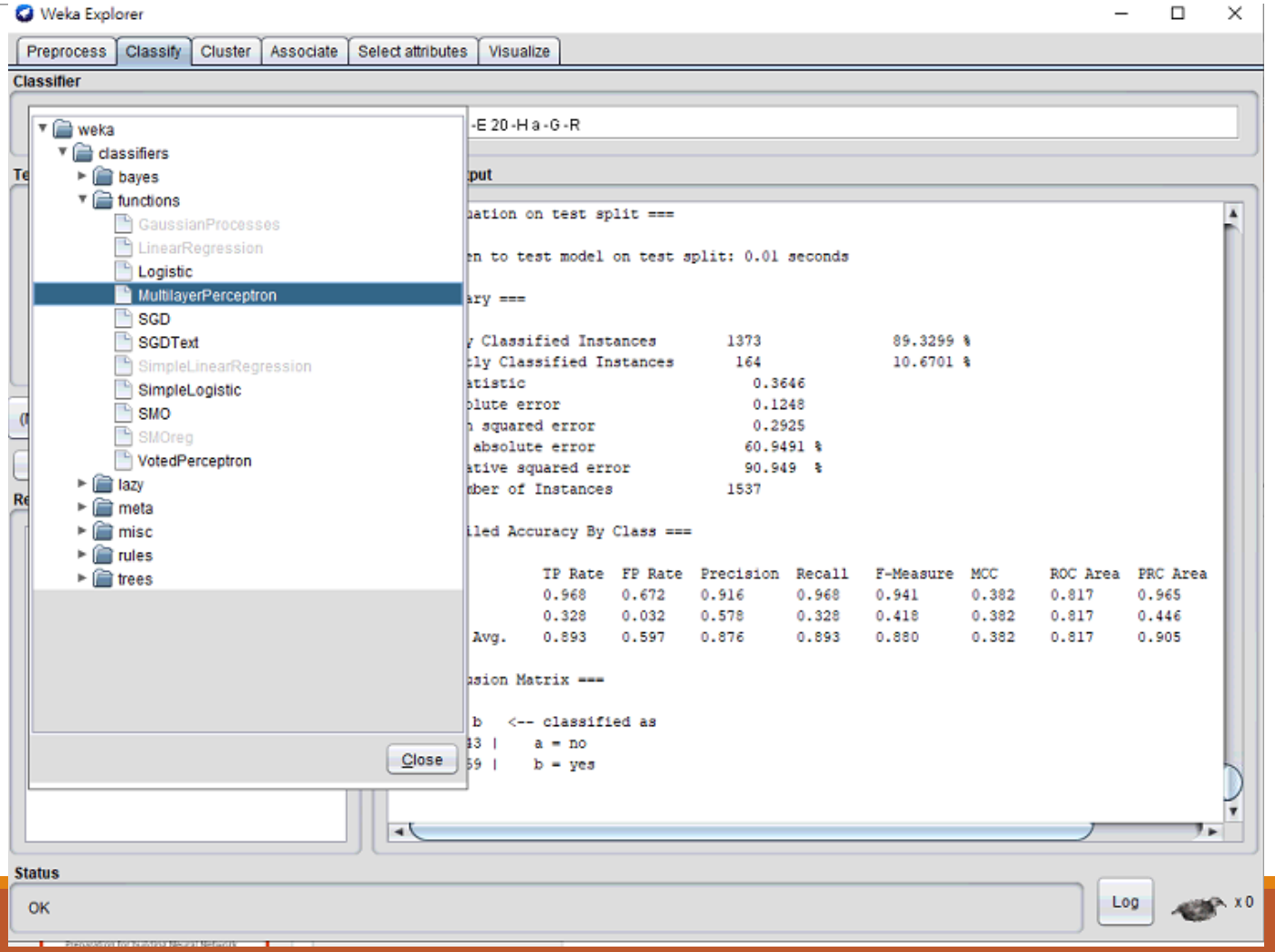


Building Neural Network

Under

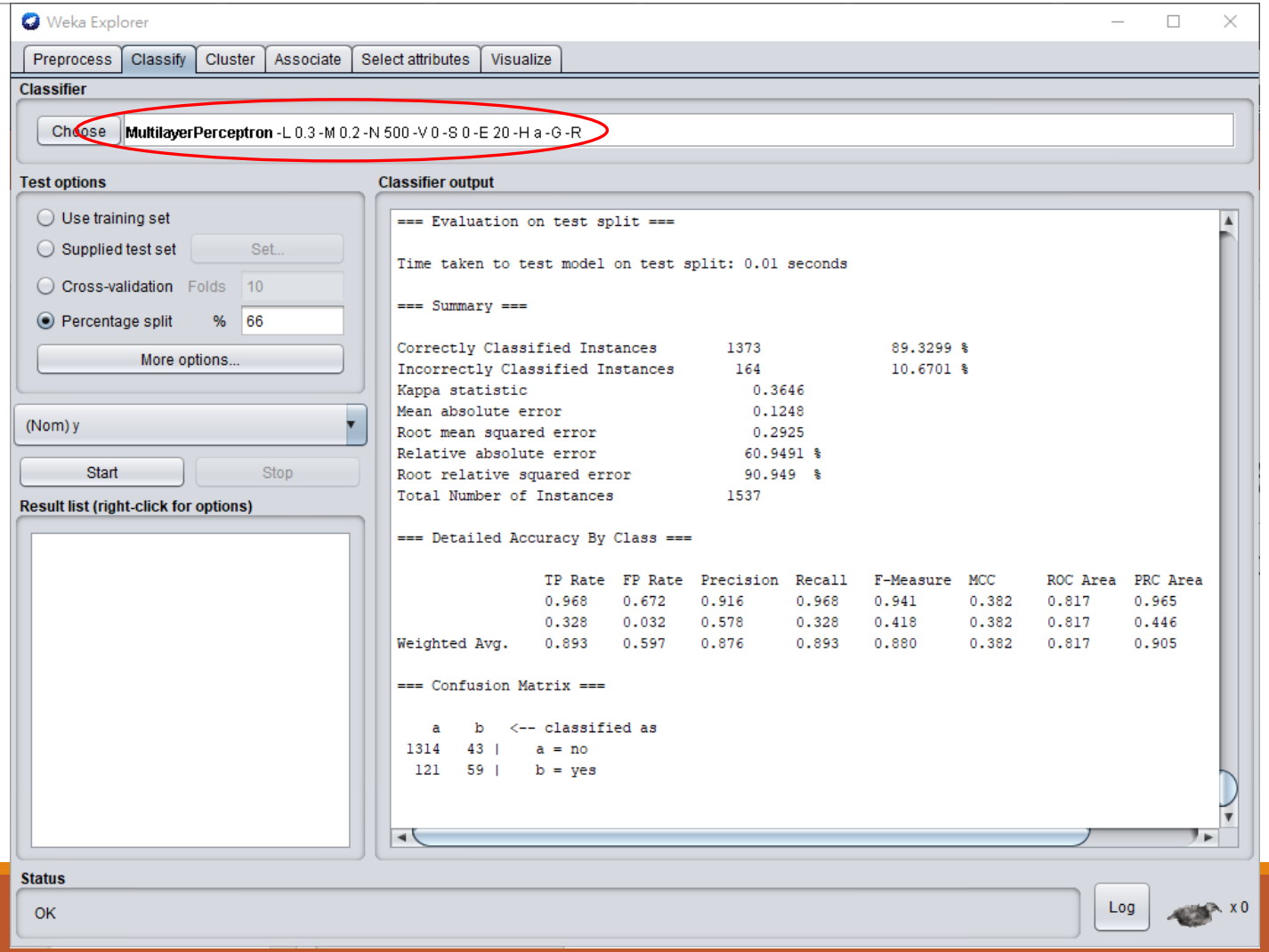
classifiers->functions

select MultilayerPerceptron



Building Neural Network

Click on the text near Choose to access to the configuration



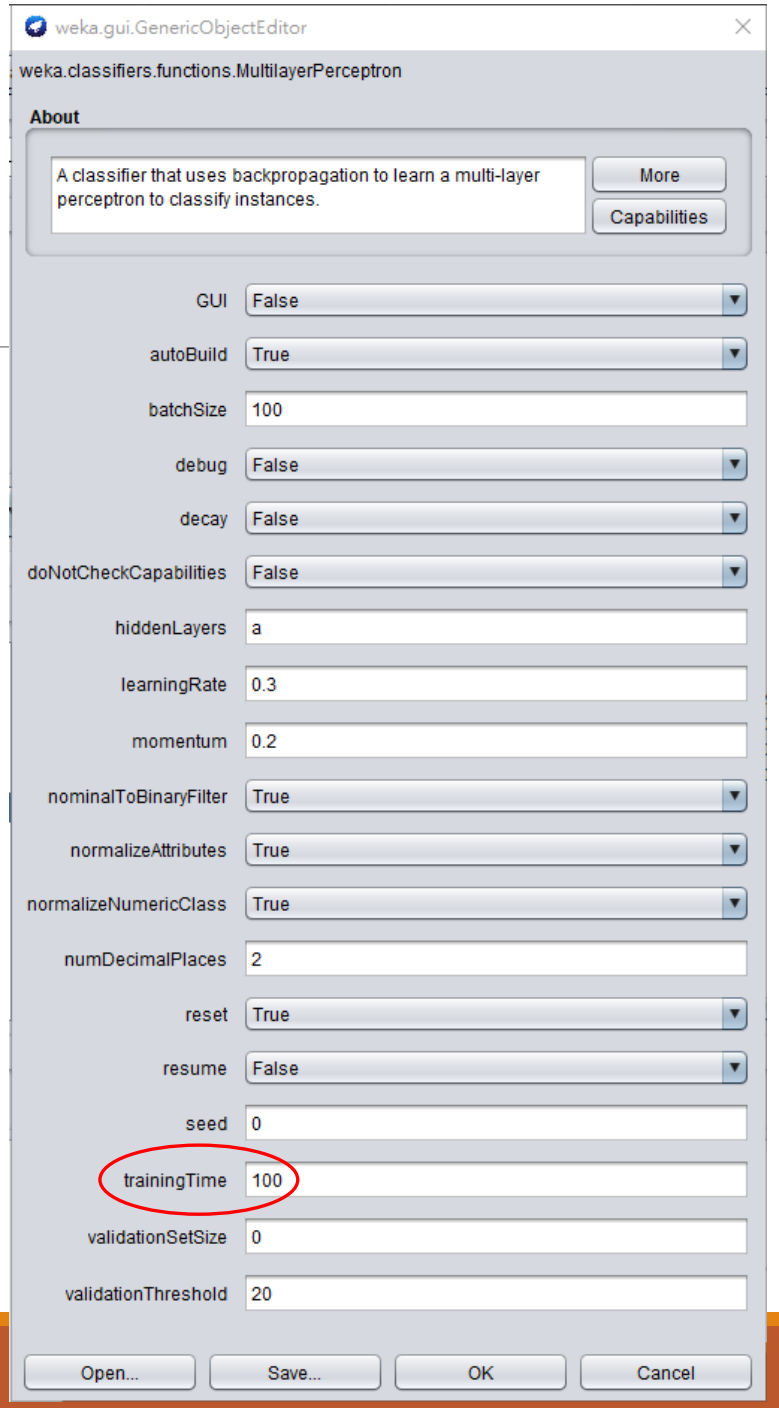
Building Neural Network

Here is the configuration of Multilayer Perceptron.

The default value of **HiddenLayers** is “a” which means Weka will help you to setup the hidden layers. You can also specify how many layer and how many nodes of each hidden layer. For example, type in 10,5,2 means 3 hidden layers with 10, 5, 2 nodes respectively.

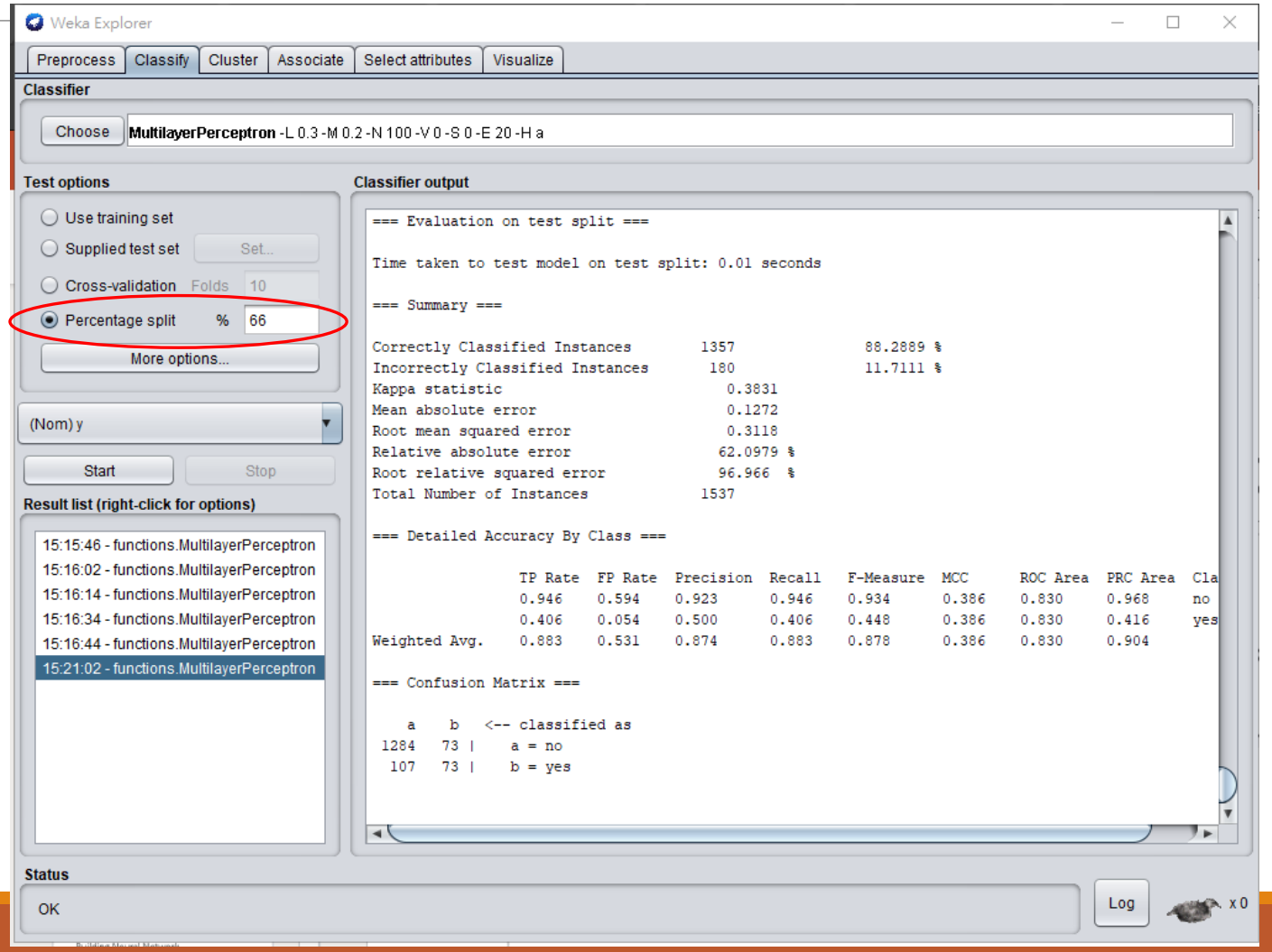
trainingTime means how many iterations we want to train through. Let set it from 500 to 100.

Then, click OK



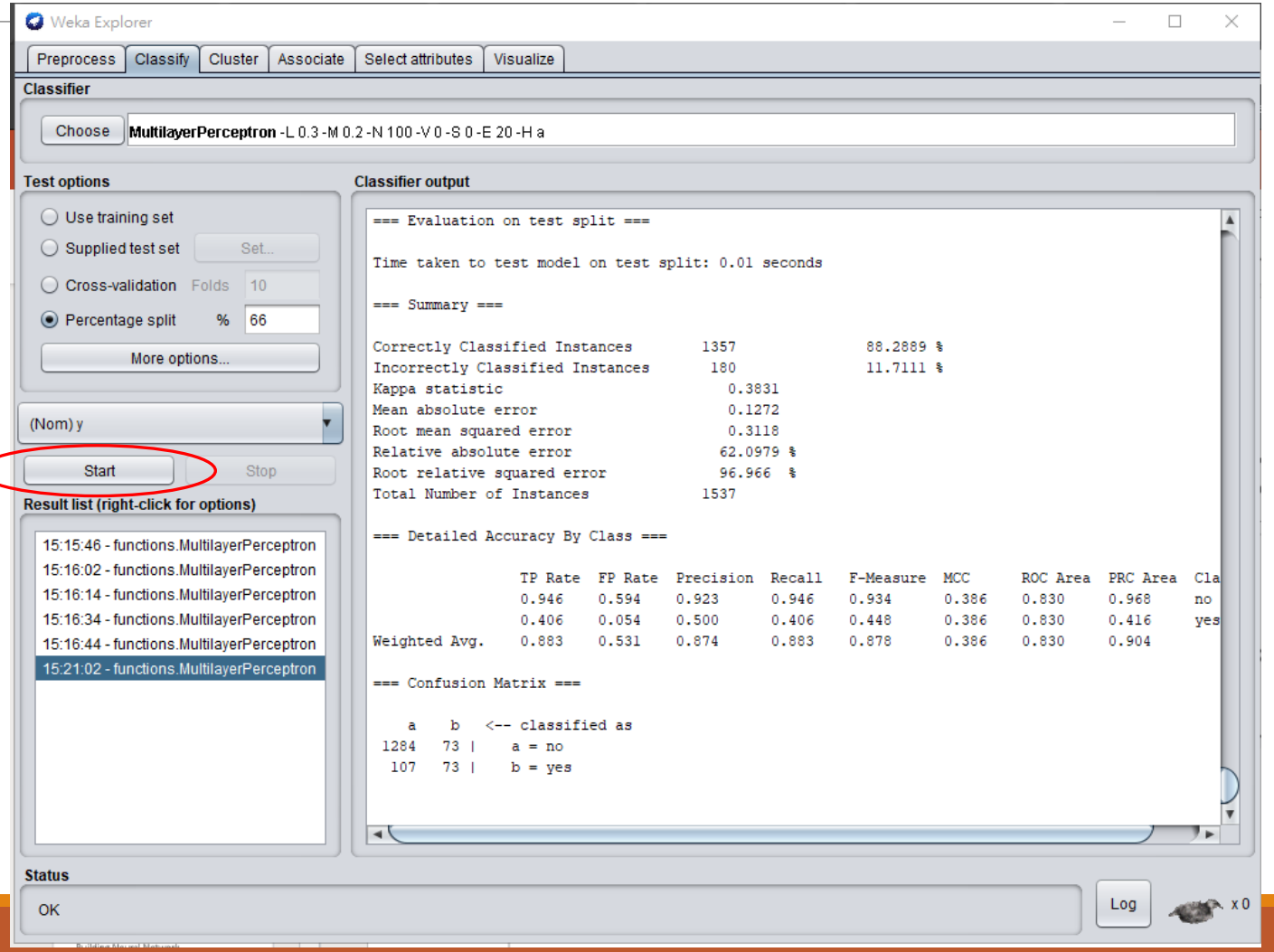
Building Neural Network

In the Test options here, we simply use percentage split 66% as our testing option.



Building Neural Network

Click Start to start our neural network construction



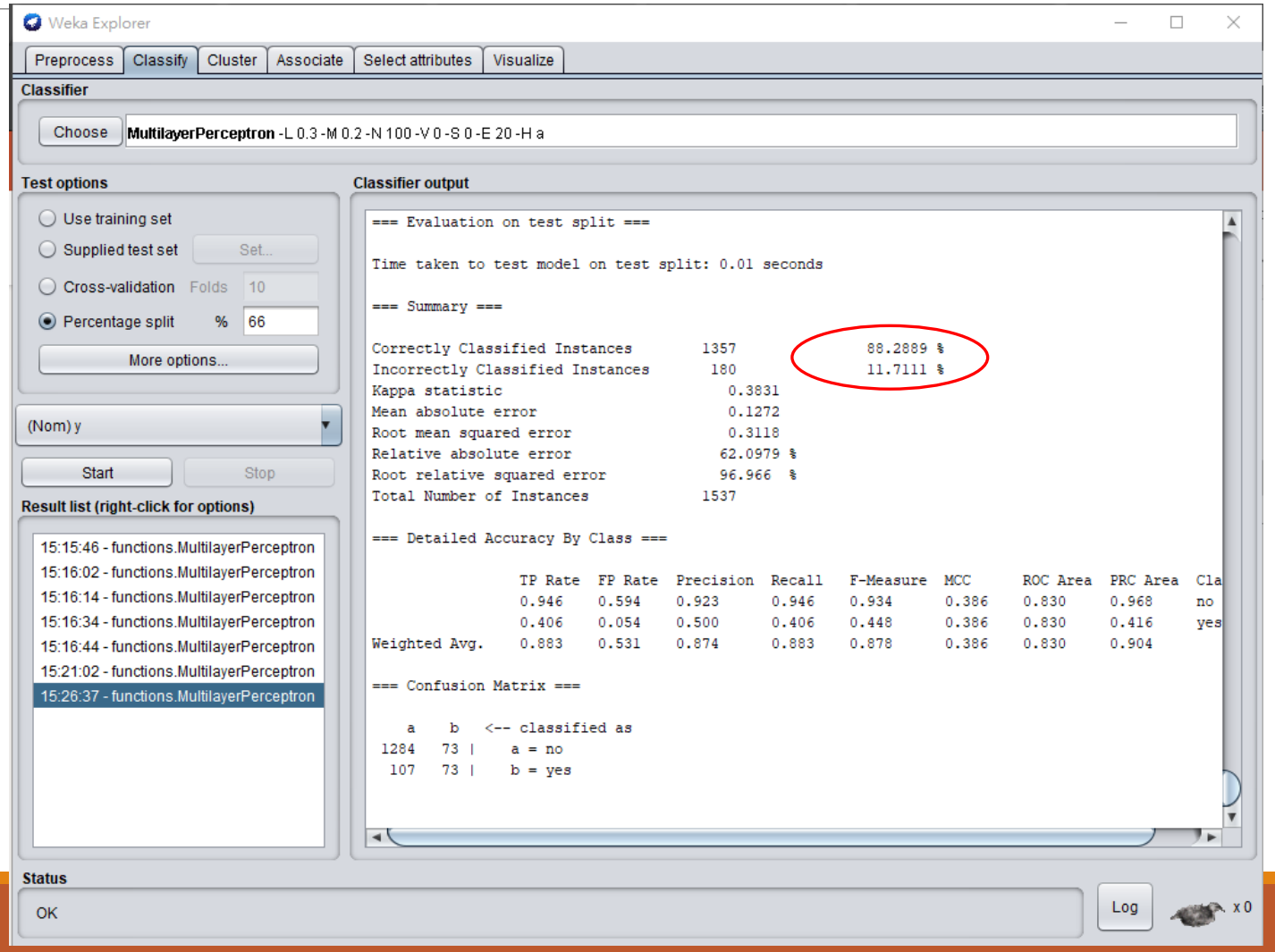
Building Neural Network

Since neural network requires much more computation power compared with decision tree and logistic regression. We need to wait Weka to train our model. The training time depends on the number of parameters (number of layers and number of nodes in each layer), number of iterations and number of data we have.

Viewing the Classifier output

After the training is finished. The result is shown on the right panel.

The accuracy of our model is 88.2889%

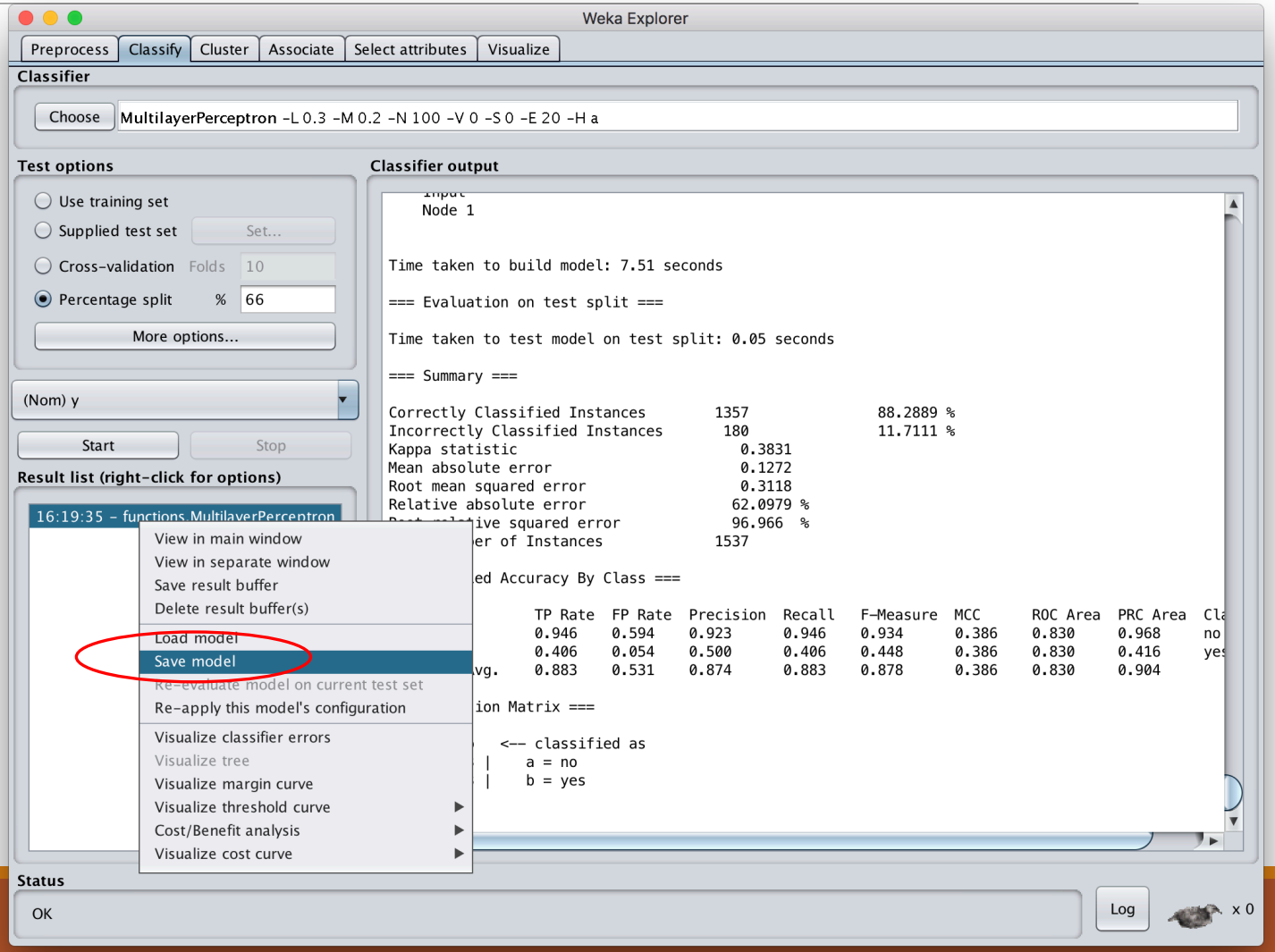


Save Neural Network Model

Suppose we want to save the trained multilayer perceptron model.

In the result list, right click the model

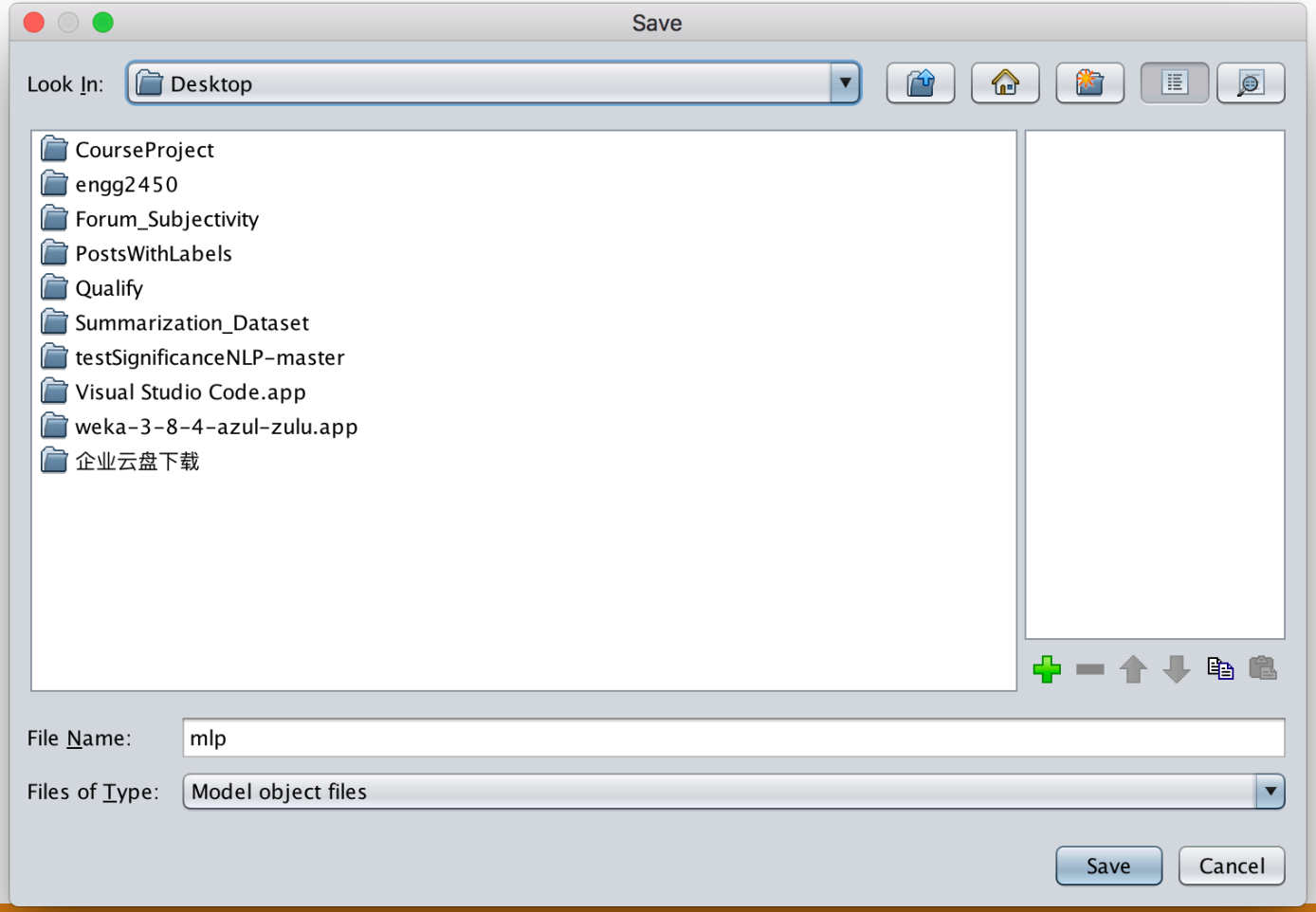
Click Save model



Save Neural Network Model

Select a location and enter a filename such as mlp, click

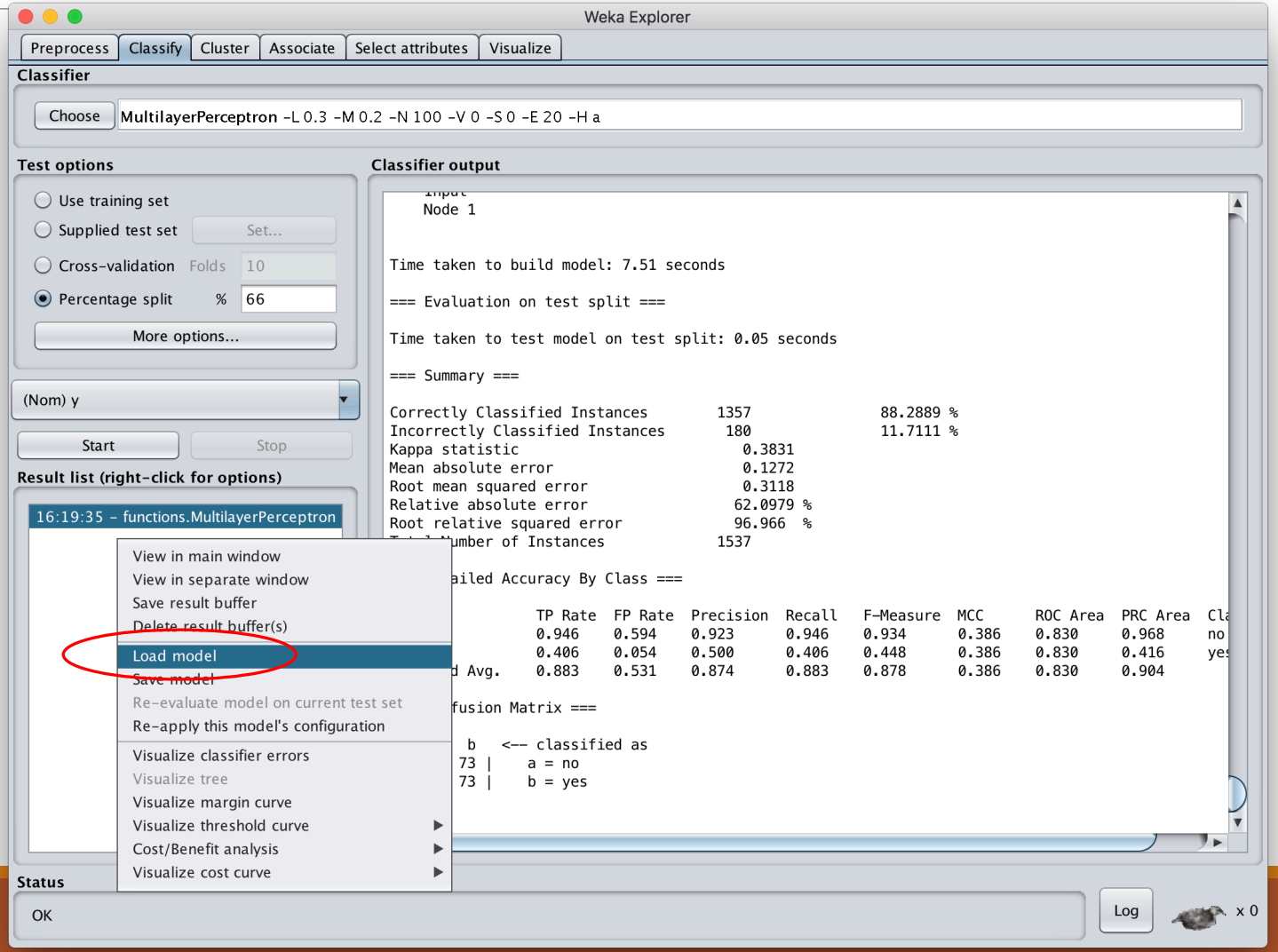
Save Our model is now saved to the file ”mlp.model".



Load Neural Network Model

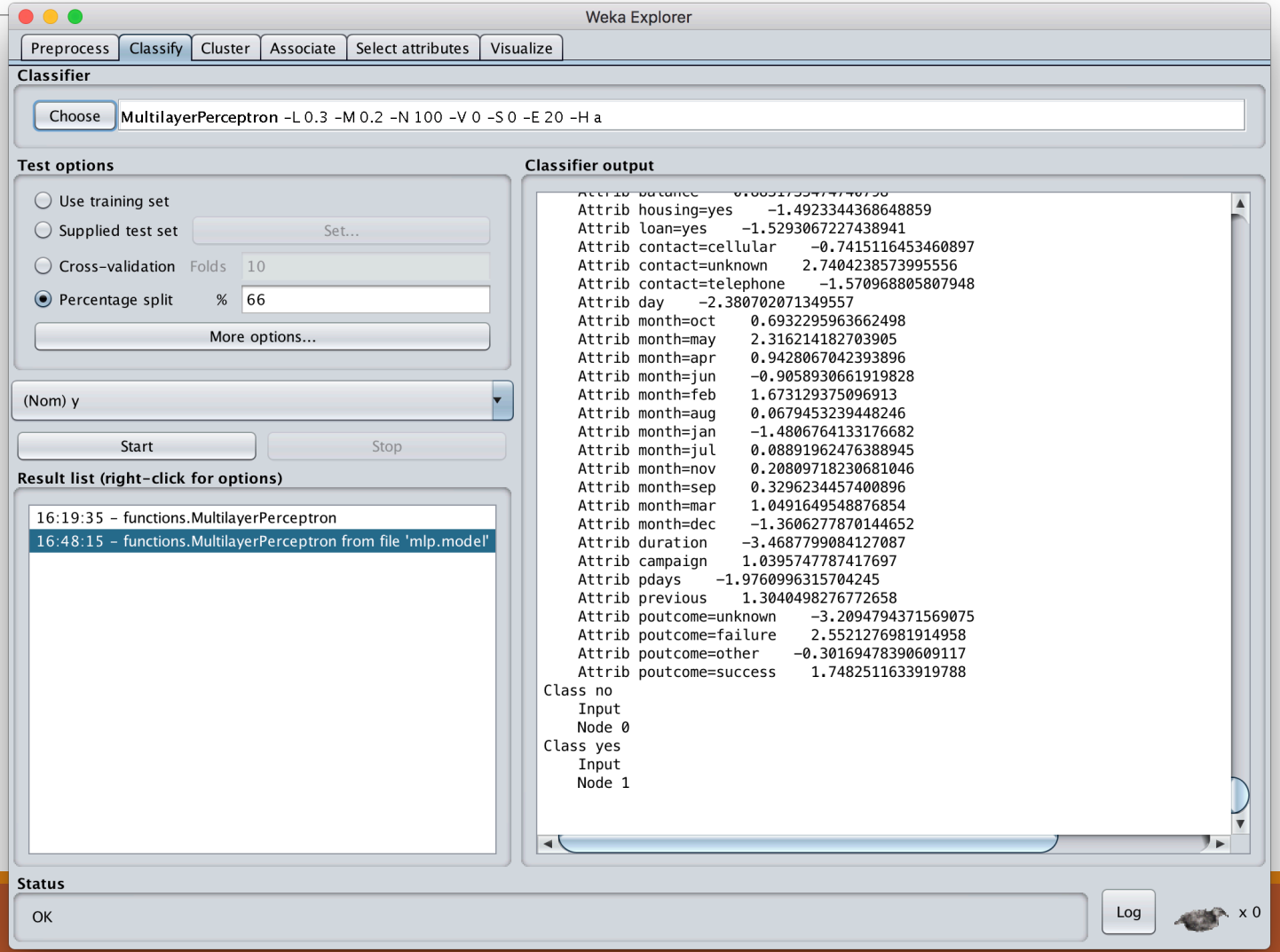
Suppose we want to use our trained model to make prediction.

Right click on the Result list and click Load model, select the model saved in the previous slide ”mlp.model".



Load Neural Network Model

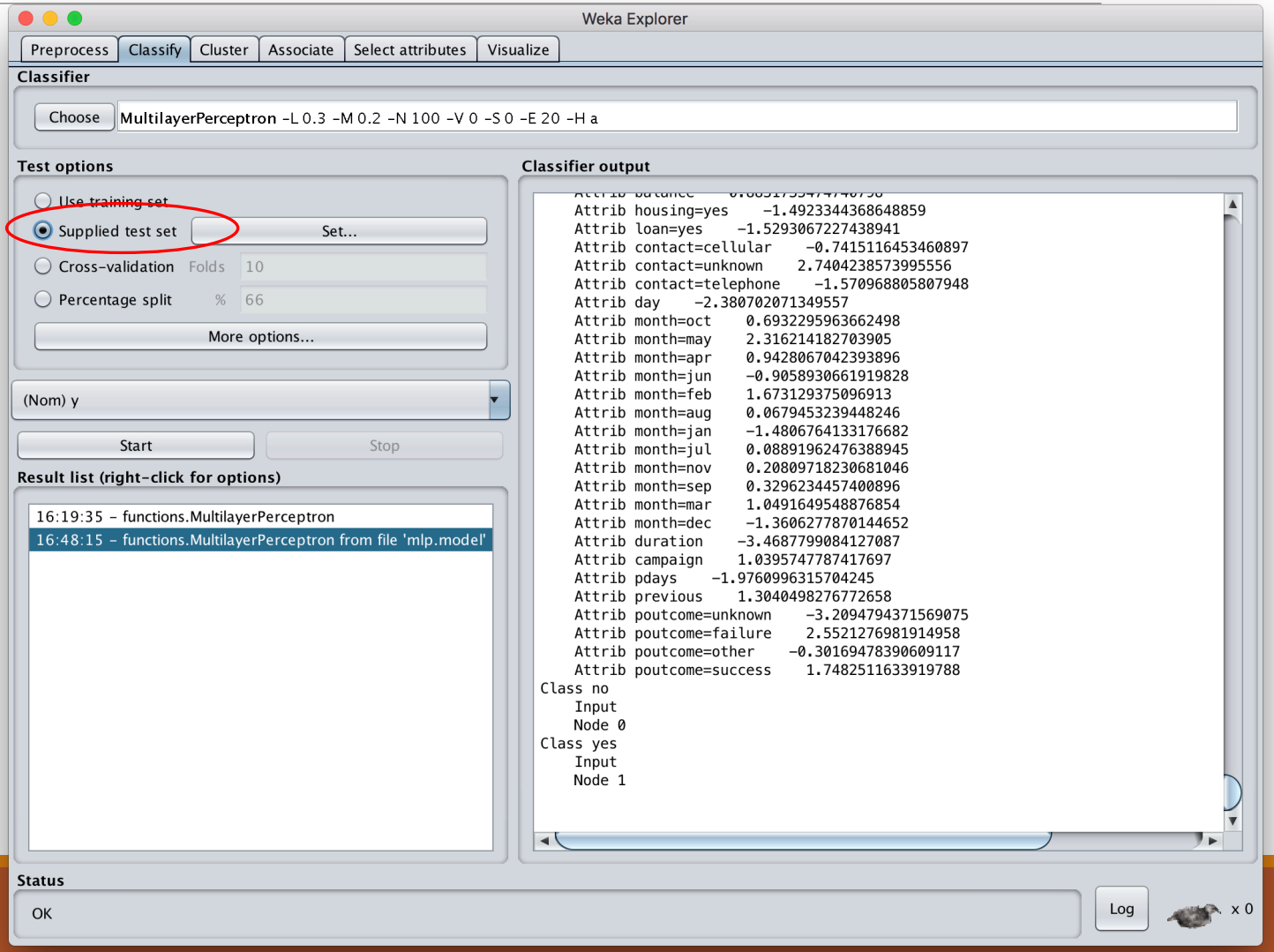
Now, the model is loaded, and we can see some information on the right panel.



Evaluate Model on New Data

We want to evaluate our model on a new dataset.

Select the Supplied test set option in the Test options pane.

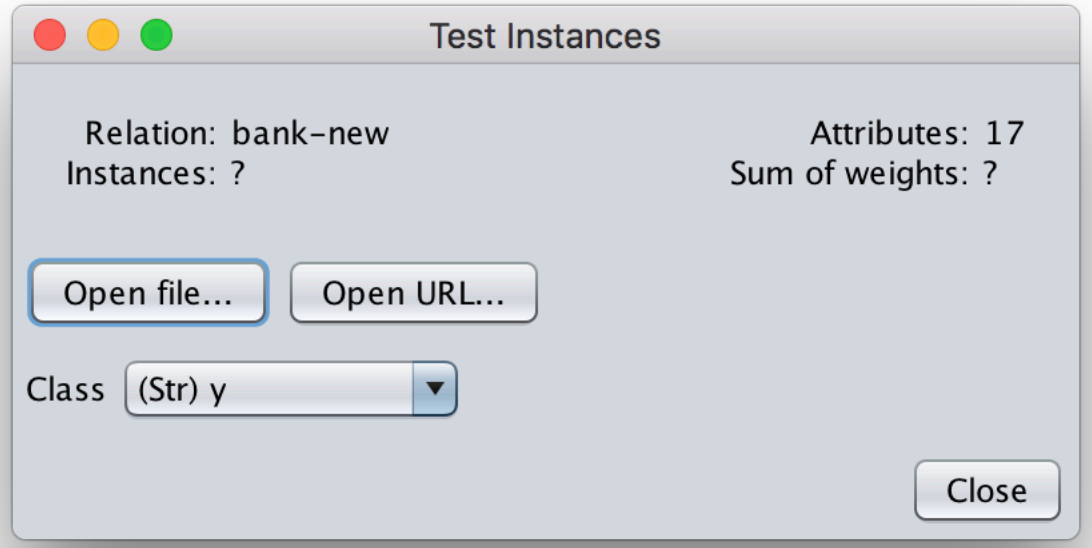


Evaluate Model on New Data

Click Set, click the Open file on the options window and select the new dataset we just created with the name "bank-additional-test.arff".

For the Class, select y

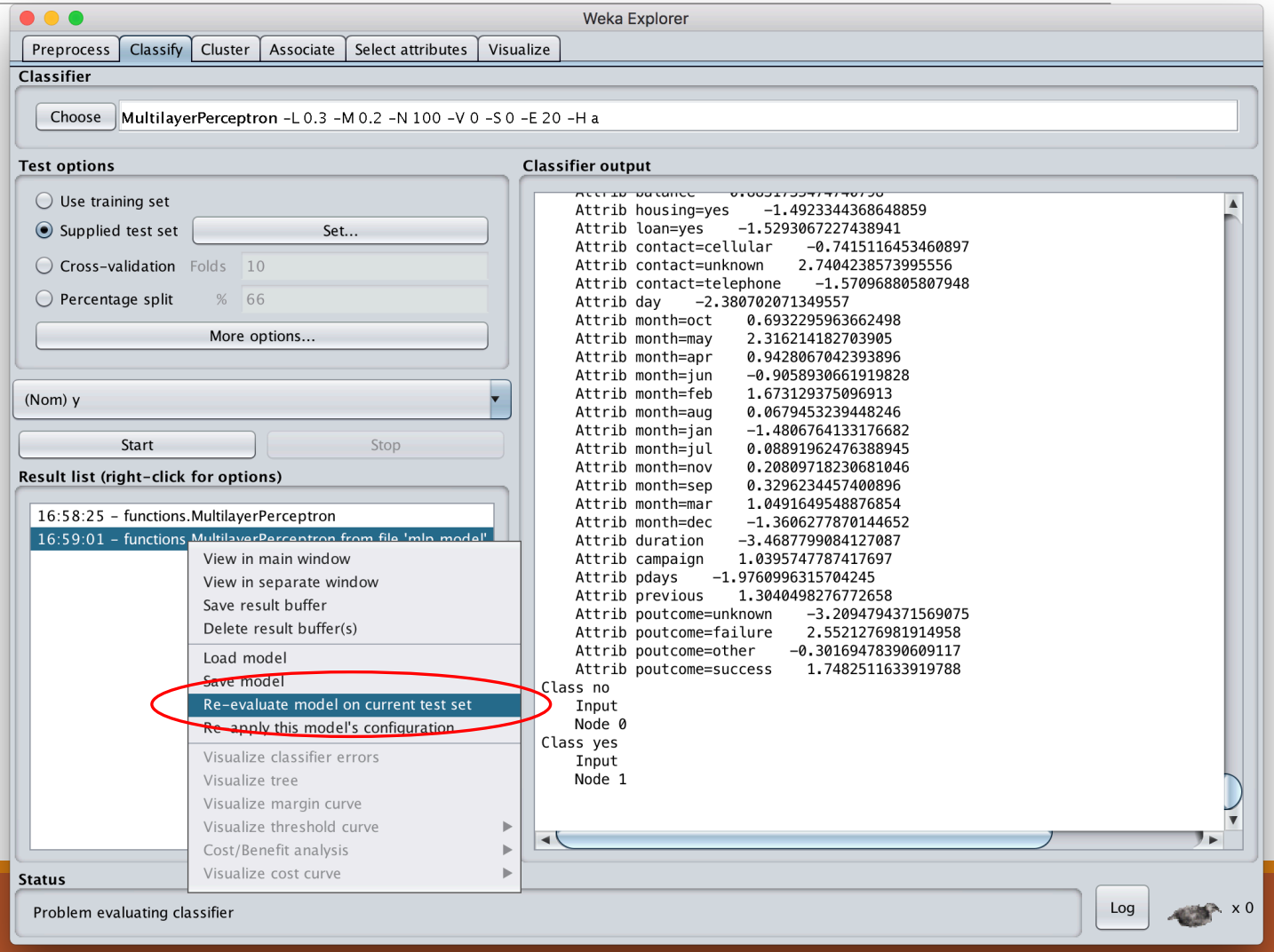
Then, Click Close



Evaluate Model on New Data

Right click on the list item for our loaded model in the Results list.

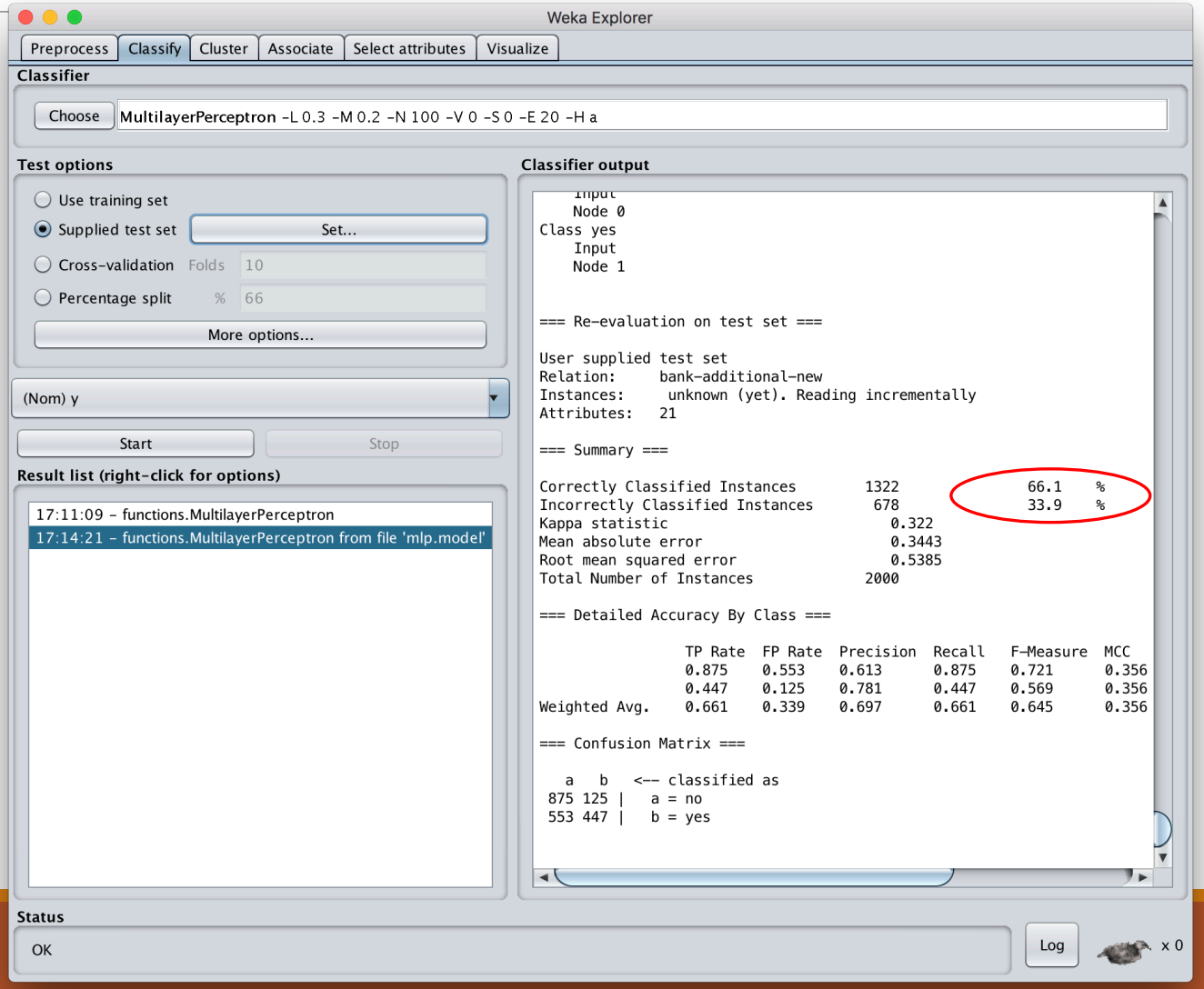
Choose Re-evaluate model on current test set



Evaluate Model on New Data

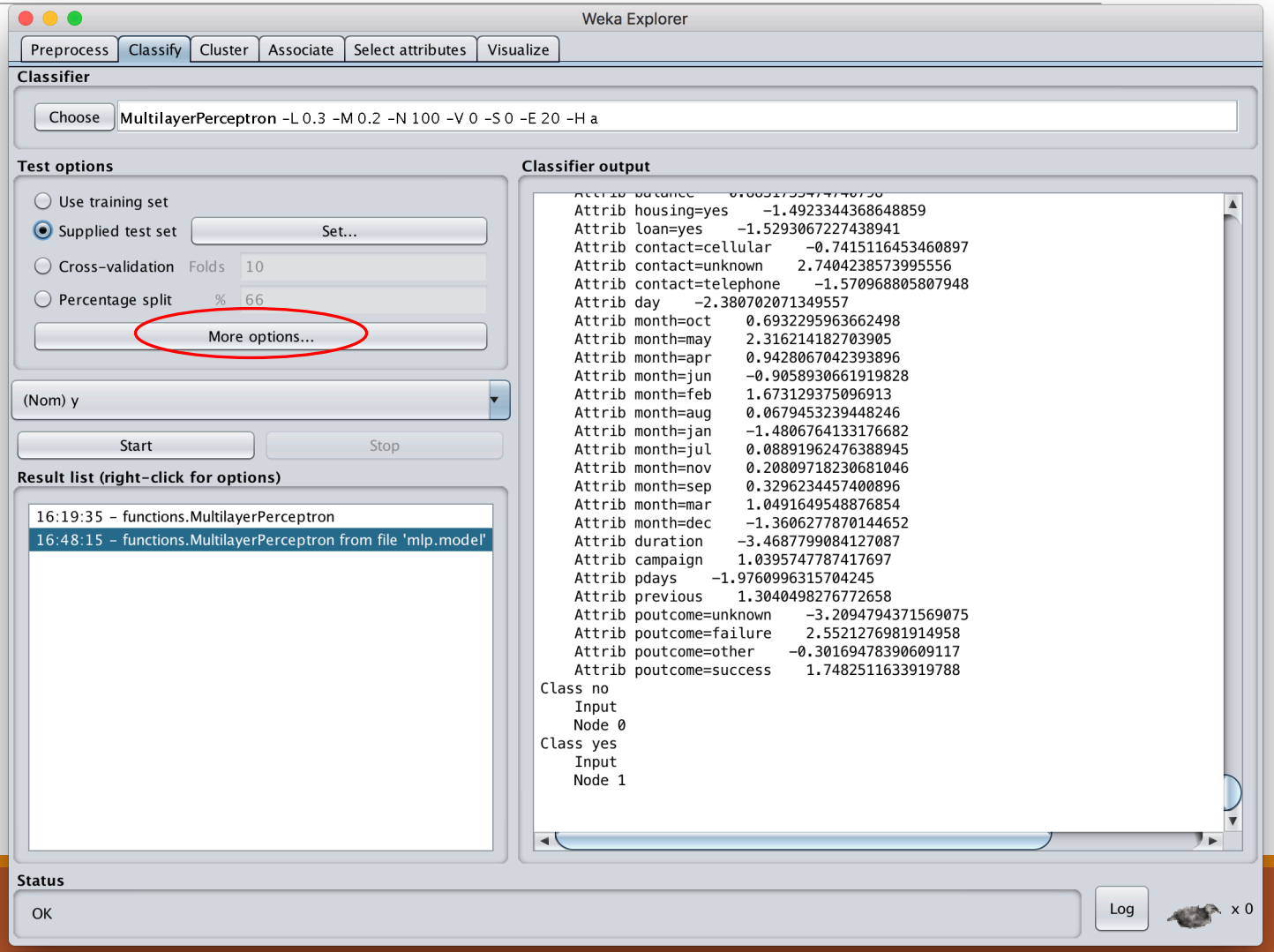
After the evaluation is finished. The result is shown on the right panel.

The accuracy of our model is 66.1%



Make Predictions on New Data

If we want to make prediction on the new data, instead of evaluate on certain metrics, click the More options… to bring up options for evaluating the classifier.



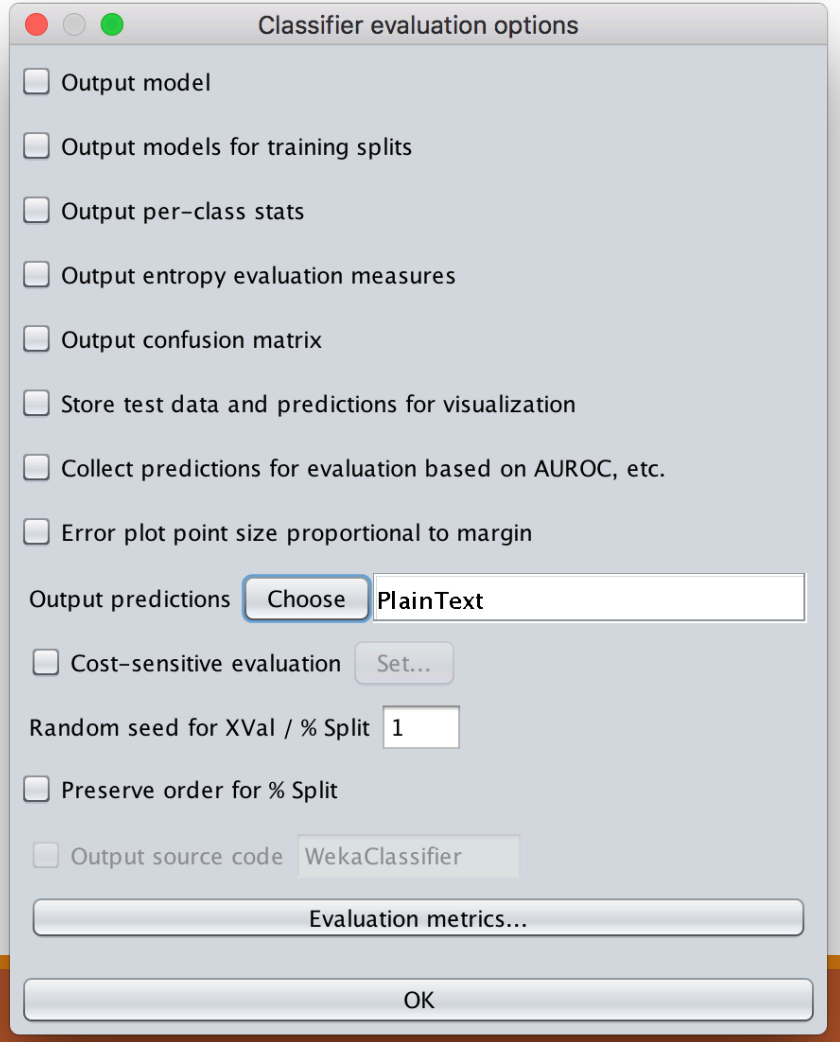
Make Predictions on New Data

Uncheck the the following information:

- Output model
- Output per-class stats
- Output confusion matrix
- Store predictions for visualization
- Collect predictions for evaluation based on AUROC, etc.

For Output predictions, choose PlainText

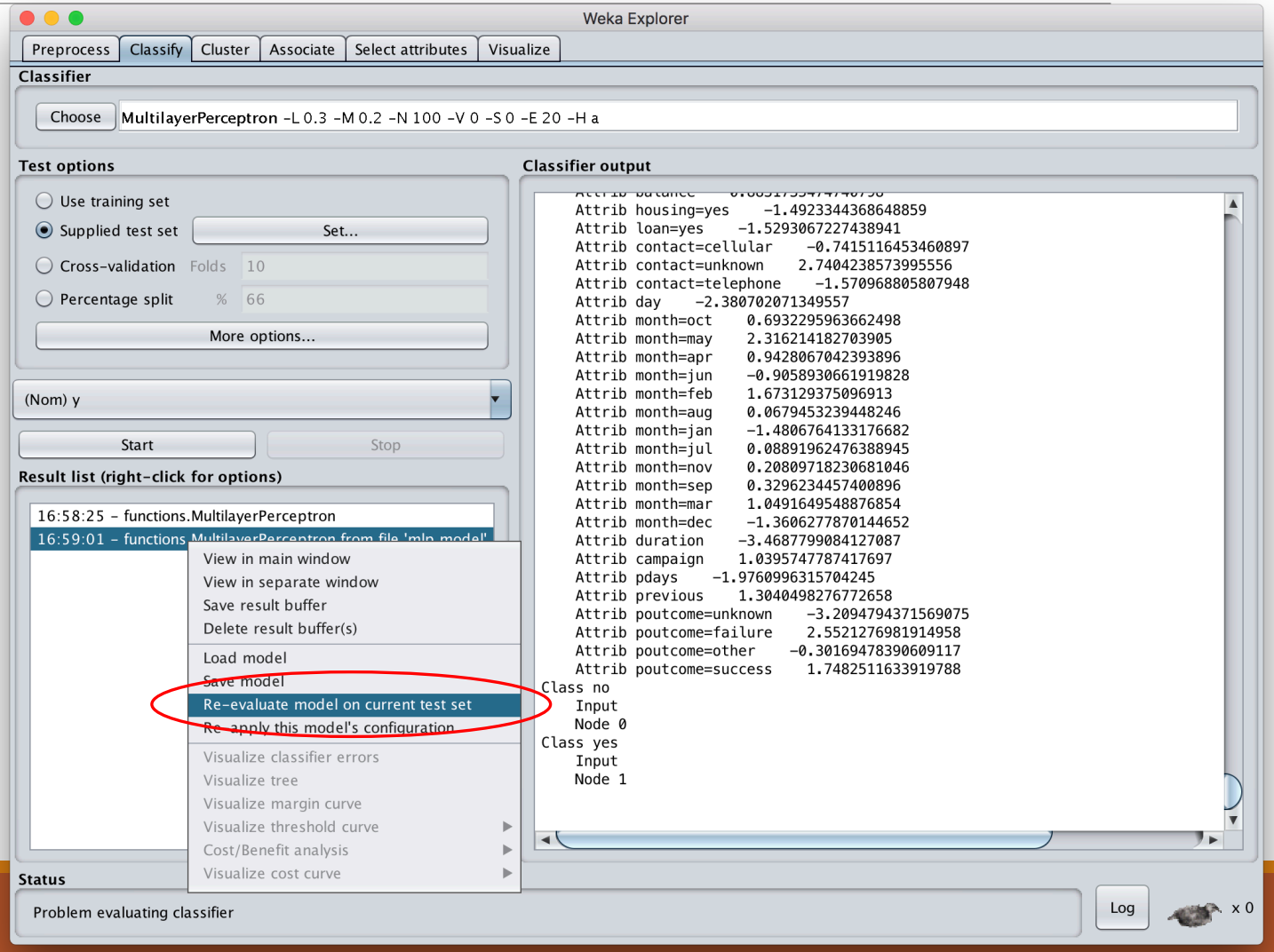
Click OK



Make Predictions on New Data

Right click on the list item for our loaded model in the Results list.

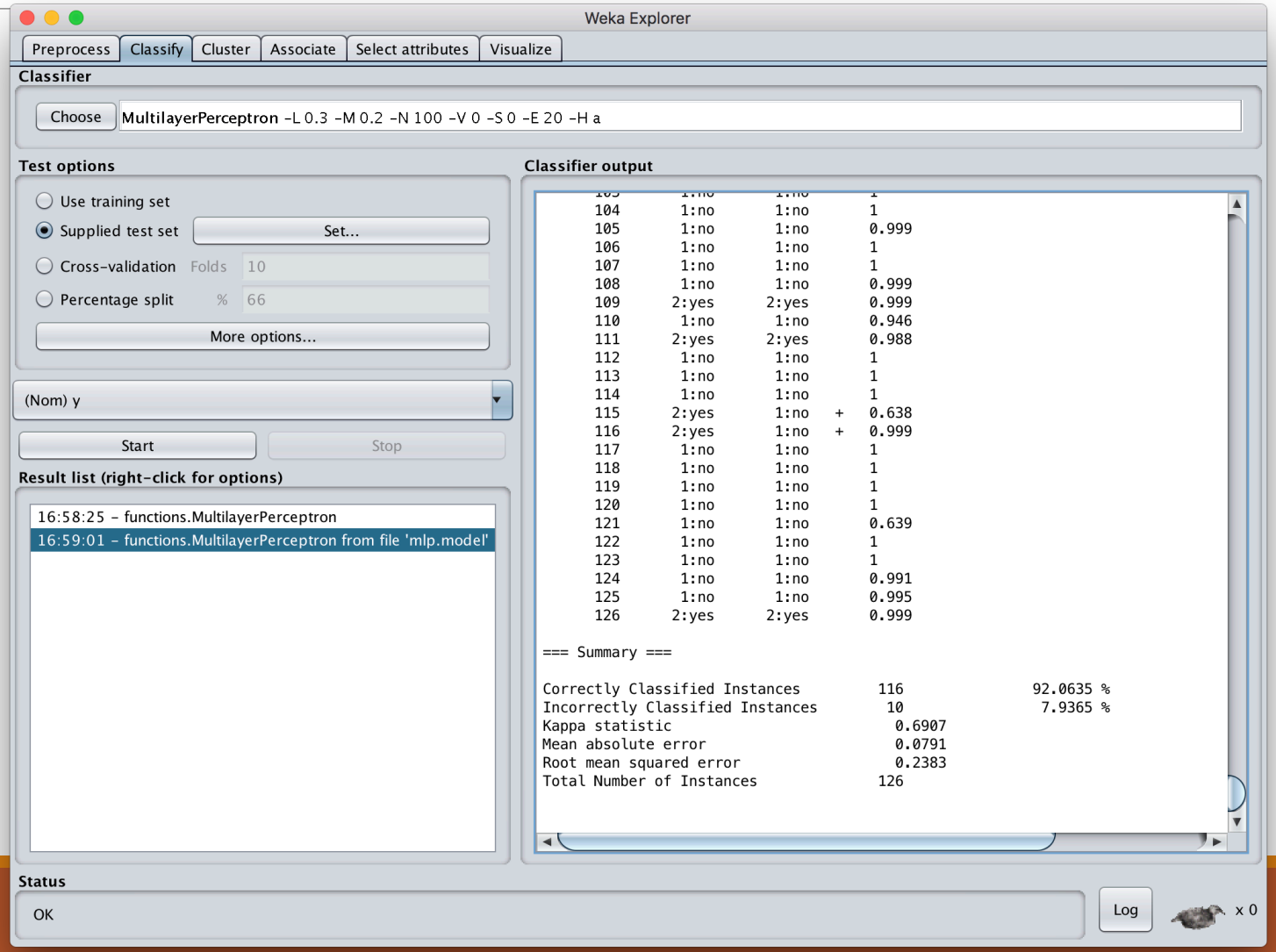
Choose Re-evaluate model again on current test set



Make Predictions on New Data

The predictions for each test instance are then listed in the Classifier Output.

Specifically, the middle column of the results is the predicted label which is "yes" or "no".

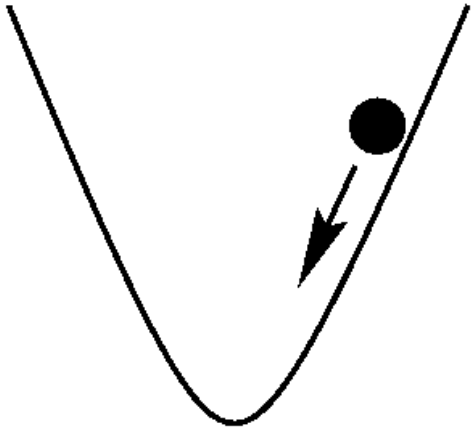


Momentum in Neural Network

In the configuration of neural network, we can see a momentum here. It is related to the optimization of the neural network. Let's briefly understand what is it in some illustrations.

Momentum in Neural Network

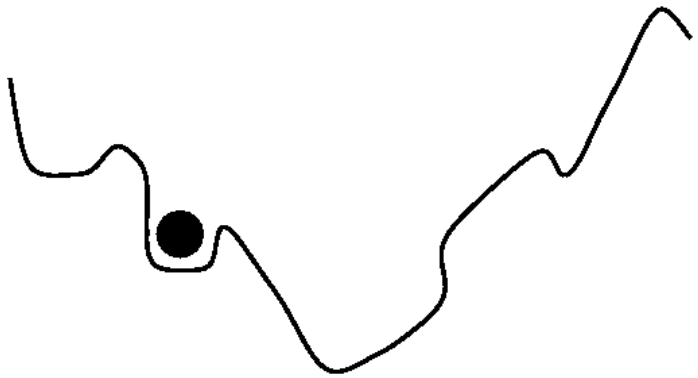
In neural networks, we use gradient descent optimization algorithm to minimize the cost function to reach a global minimum. In an ideal situation, the cost function would look like this.



So we are guaranteed to find the global optimum because there is no local minimum where the optimization can get stuck.

Momentum in Neural Network

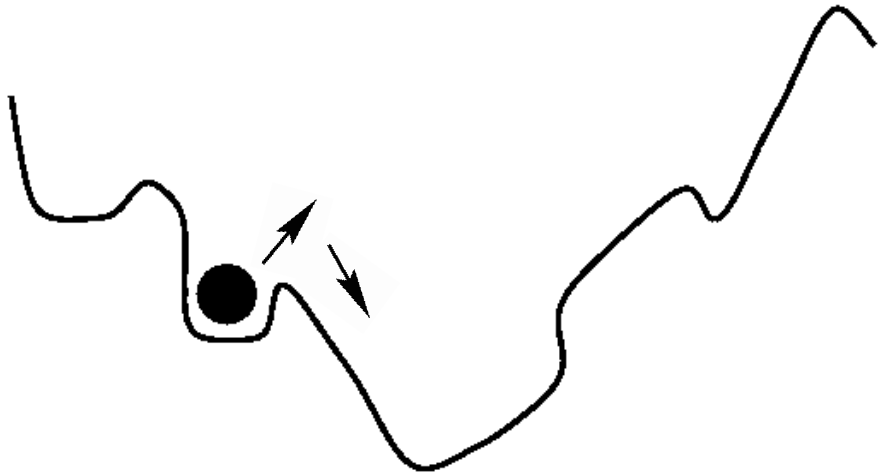
However, that is not easy. In real world, the cost function looks like more complex which comprise of several local minima and may look like this



In this case, it gets stuck in a local minimum easily and the optimization algorithm may think we already reach the global minimum which leads to sub-optimal results.

Momentum in Neural Network

To avoid this situation, we add a momentum when updating the parameters, which is a value between 0 and 1 that increases the size of the steps taken towards the minimum by trying to jump from a local minimum.



Momentum in Neural Network

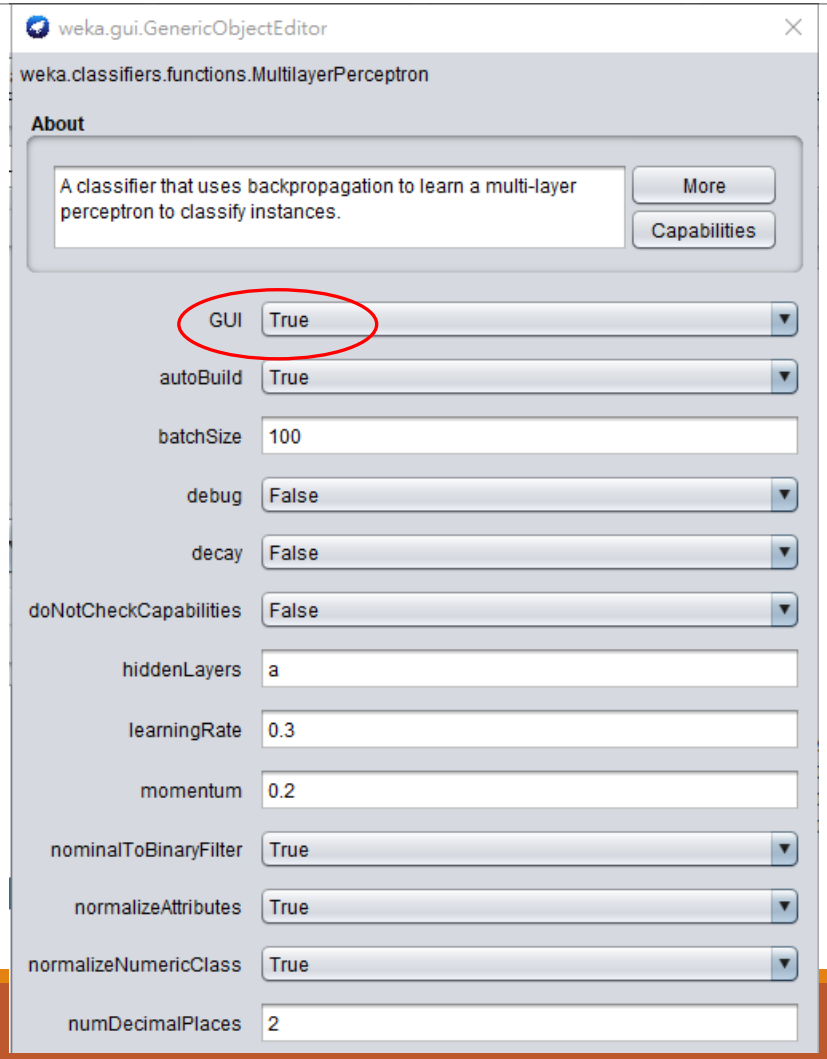
For more details about the optimization of neural network, you can find out more recourses in some optimization courses.

Building Neural Network using GUI

Let us investigate more configuration of the neural network.

Change GUI from False to True. This will provide a GUI windows after clicking Start.

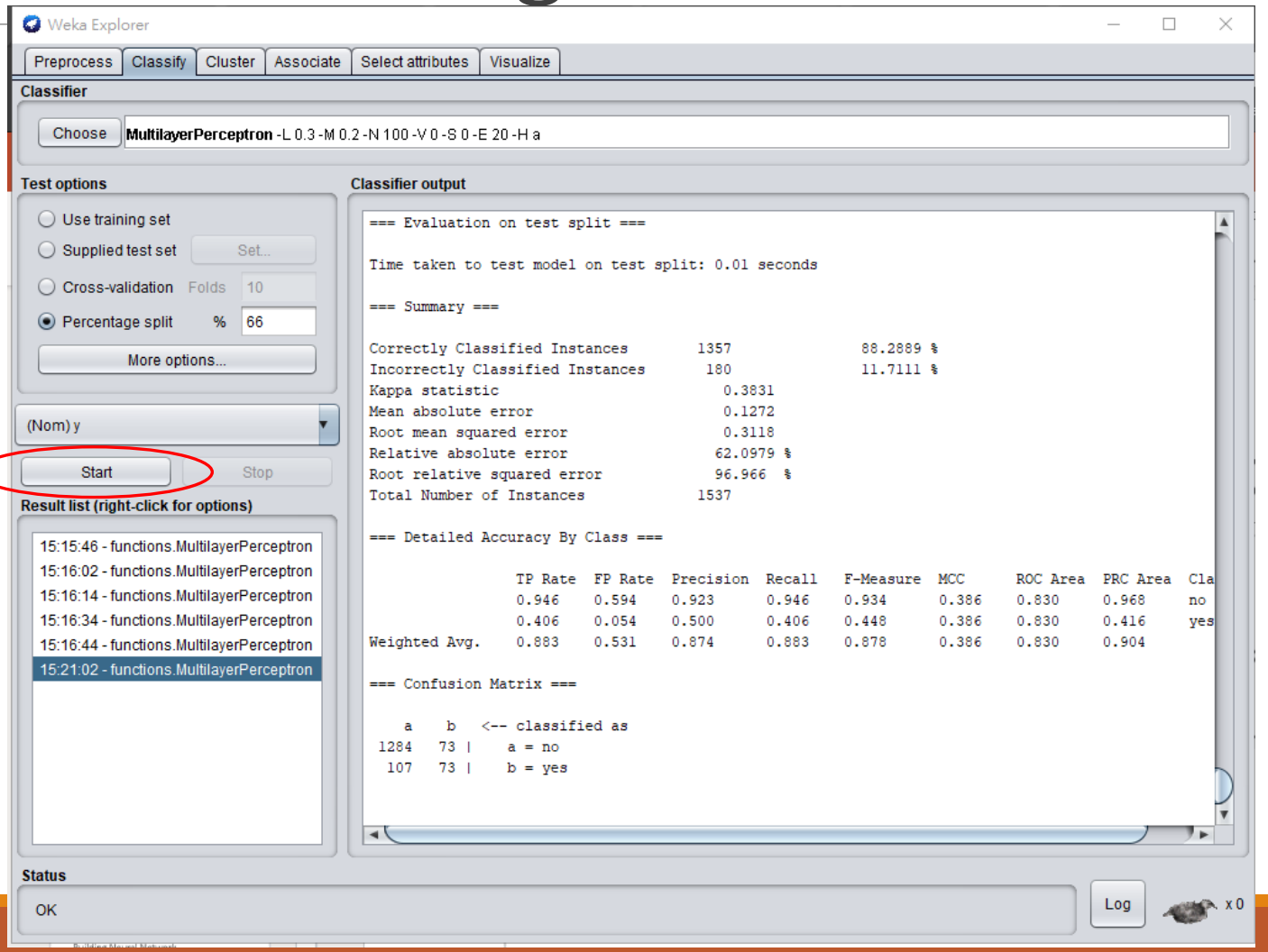
Click OK to close the configuration.



Building Neural Network using GUI

Leave other setting as the same as previous slides.

Click Start

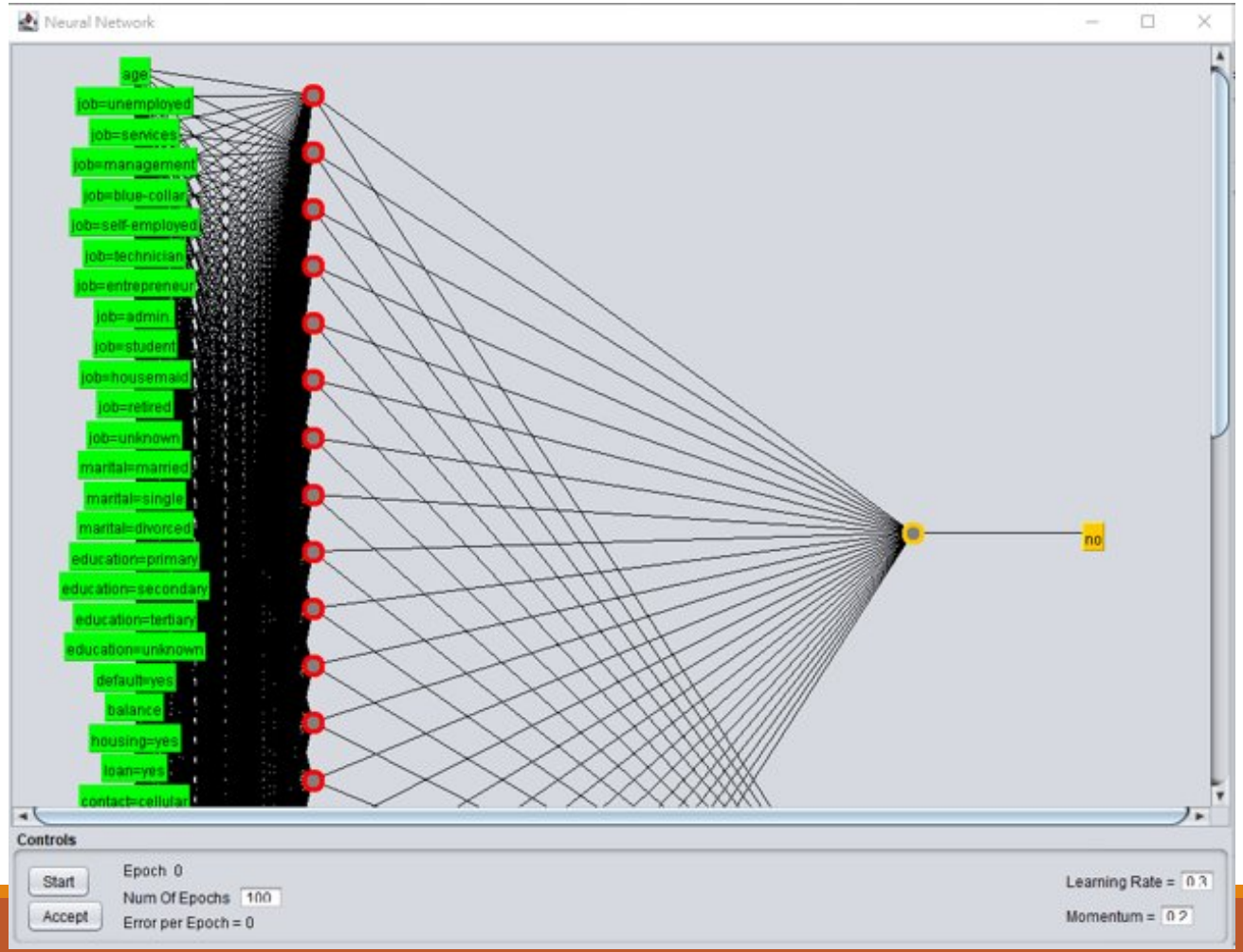


Building Neural Network using GUI

A GUI window will be shown.

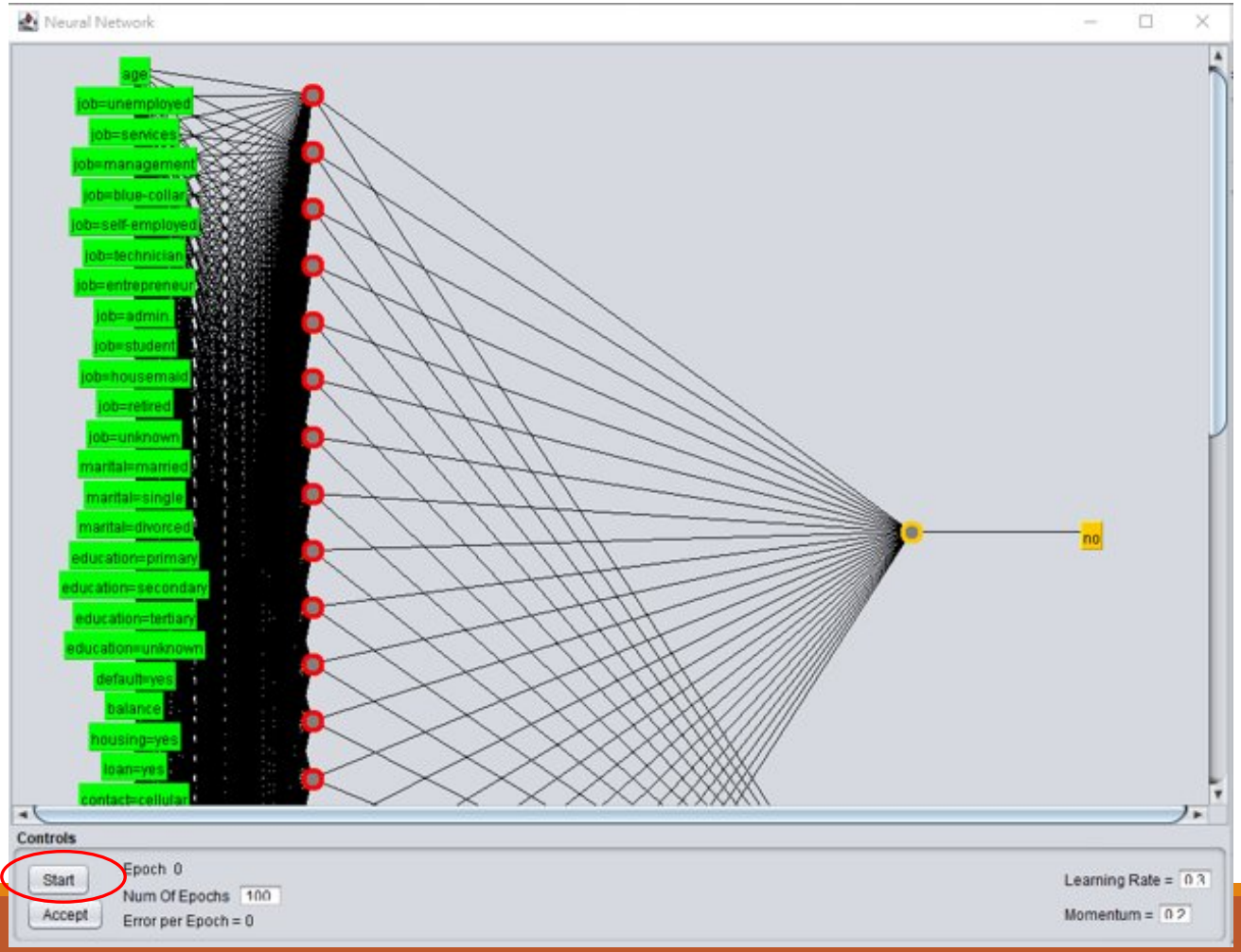
Here, you can see the structure of the neural network.

Also, you can change some of the configurations here like number of Epochs (**trainingTime**), Learning Rate and Momentum.



Building Neural Network using GUI

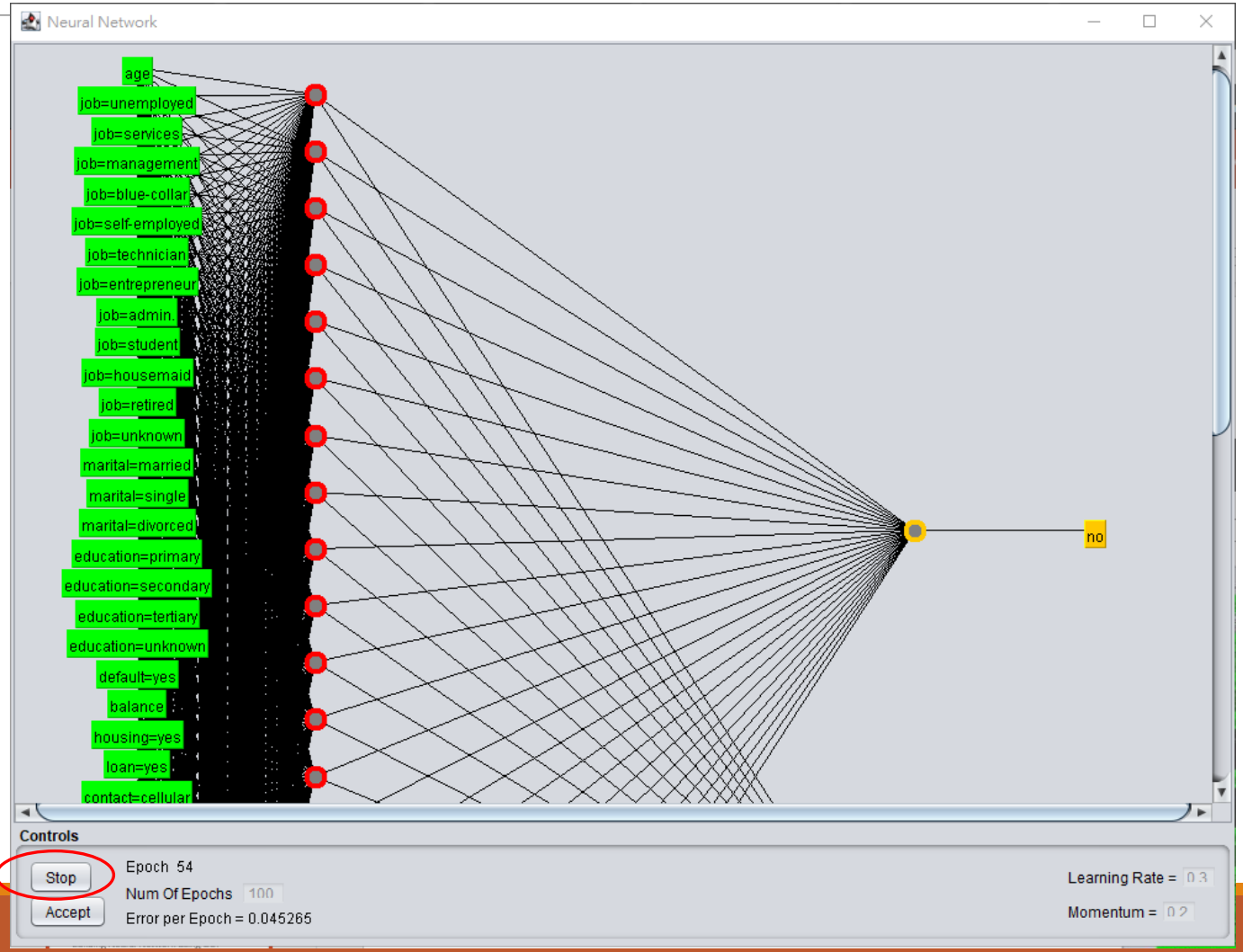
You can click Start to start training.



Building Neural Network using GUI

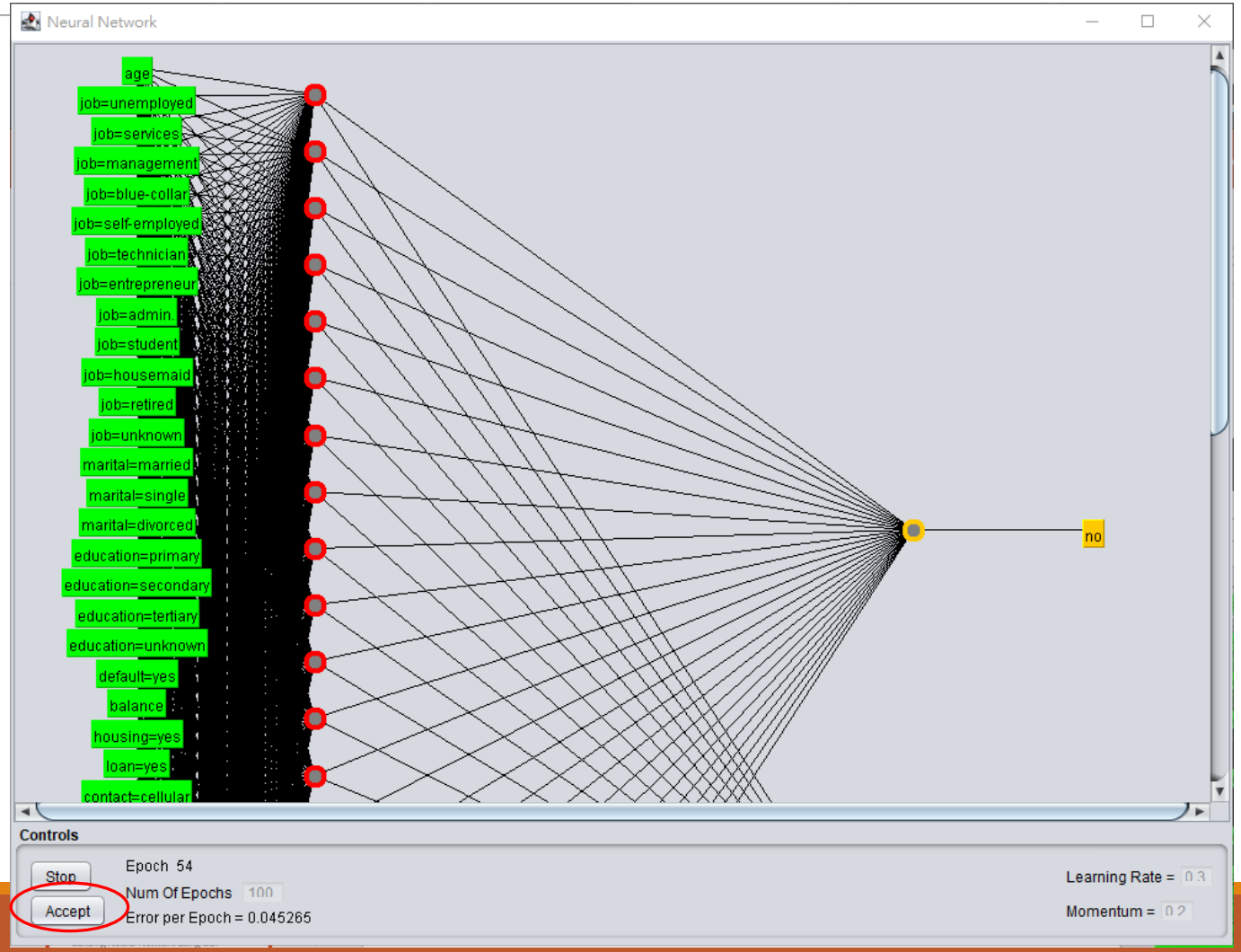
You can click Stop to stop training.

Also, the current epoch and error per epoch will be updated continuously after each epoch.



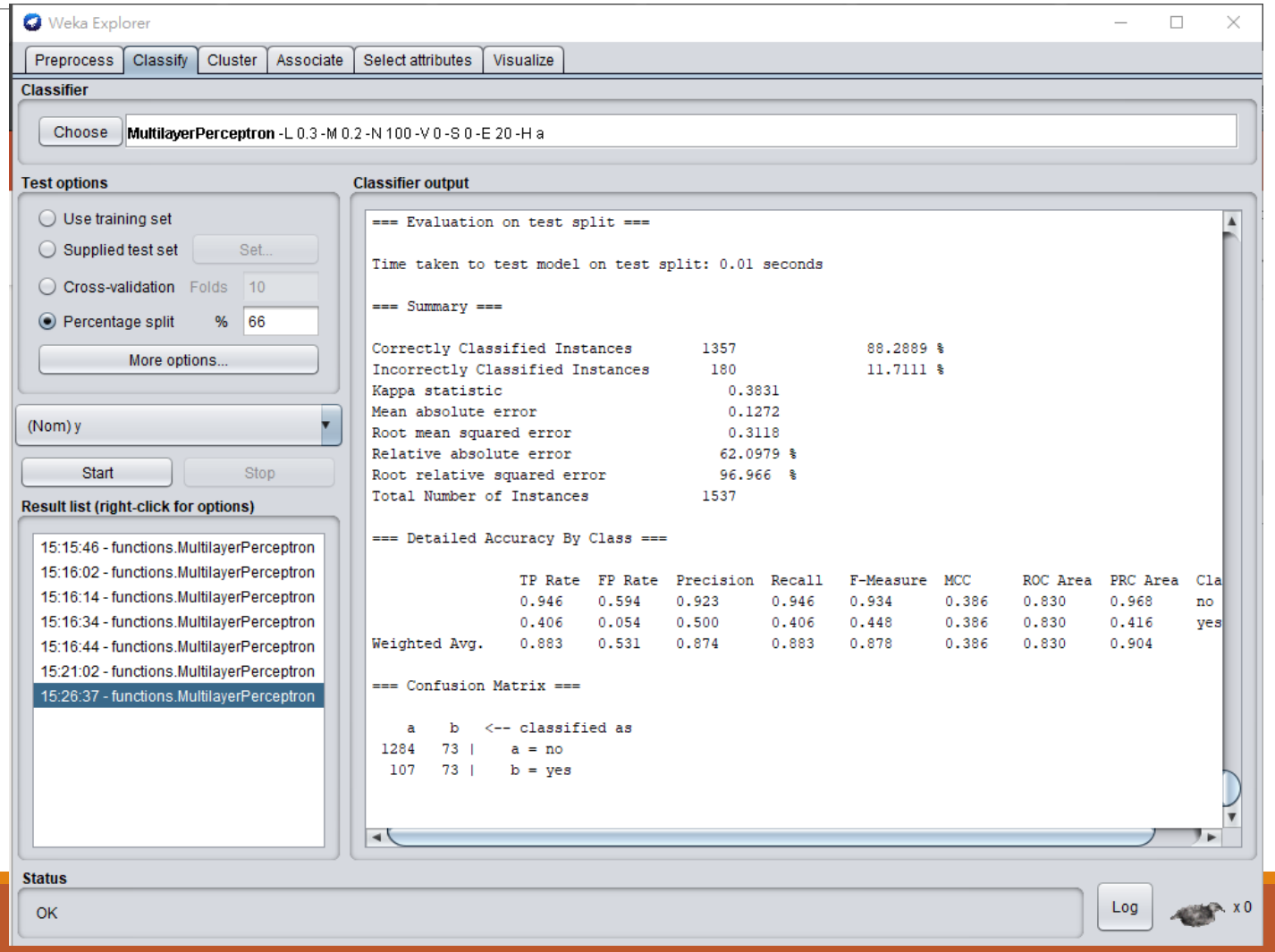
Building Neural Network using GUI

You can click Accept to finish training although the training has not reached the 100 epochs.



Building Neural Network using GUI

After you click Accept, the result will be shown on the right panel, just like previous slides.

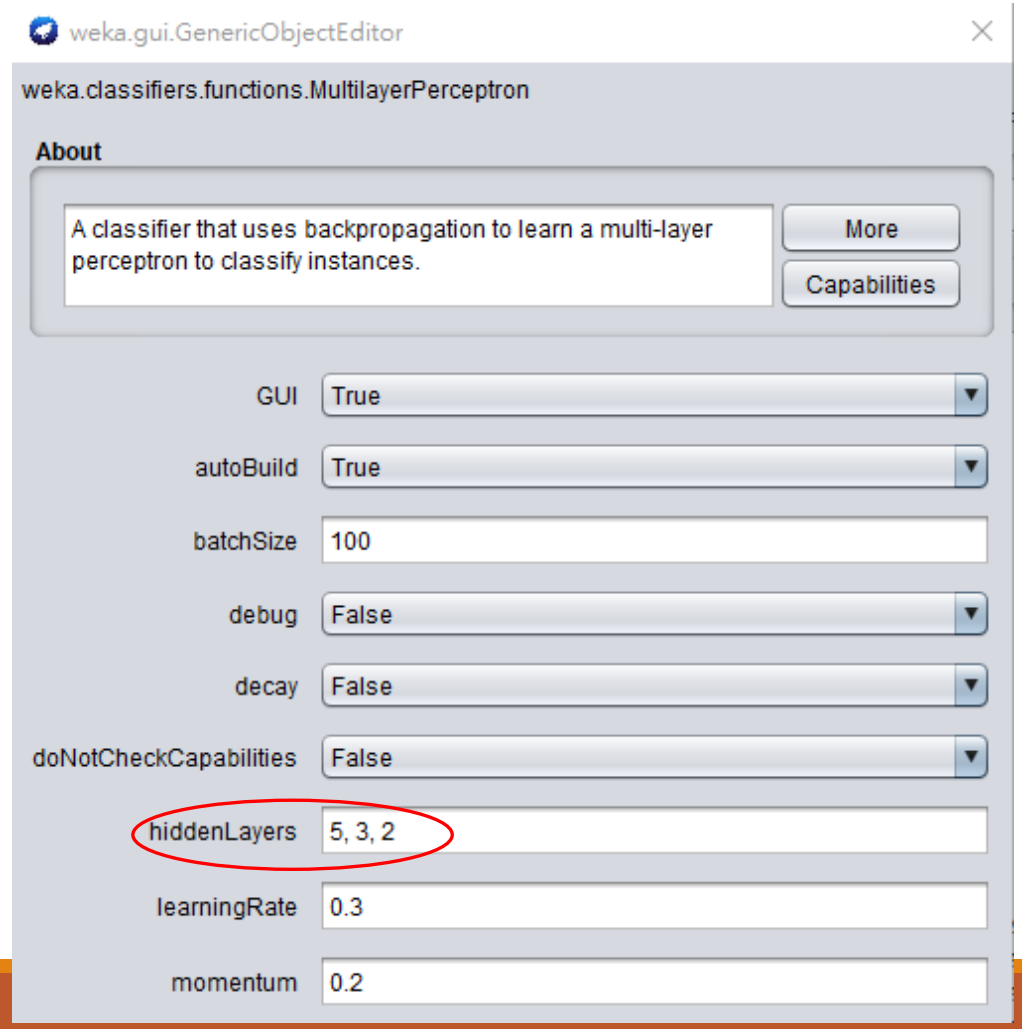


Building Neural Network using GUI

As mentioned in previous slides, you can specify the number of hidden layers and the nodes of each layer. If you turn on the GUI mode, you can easily to verify it.

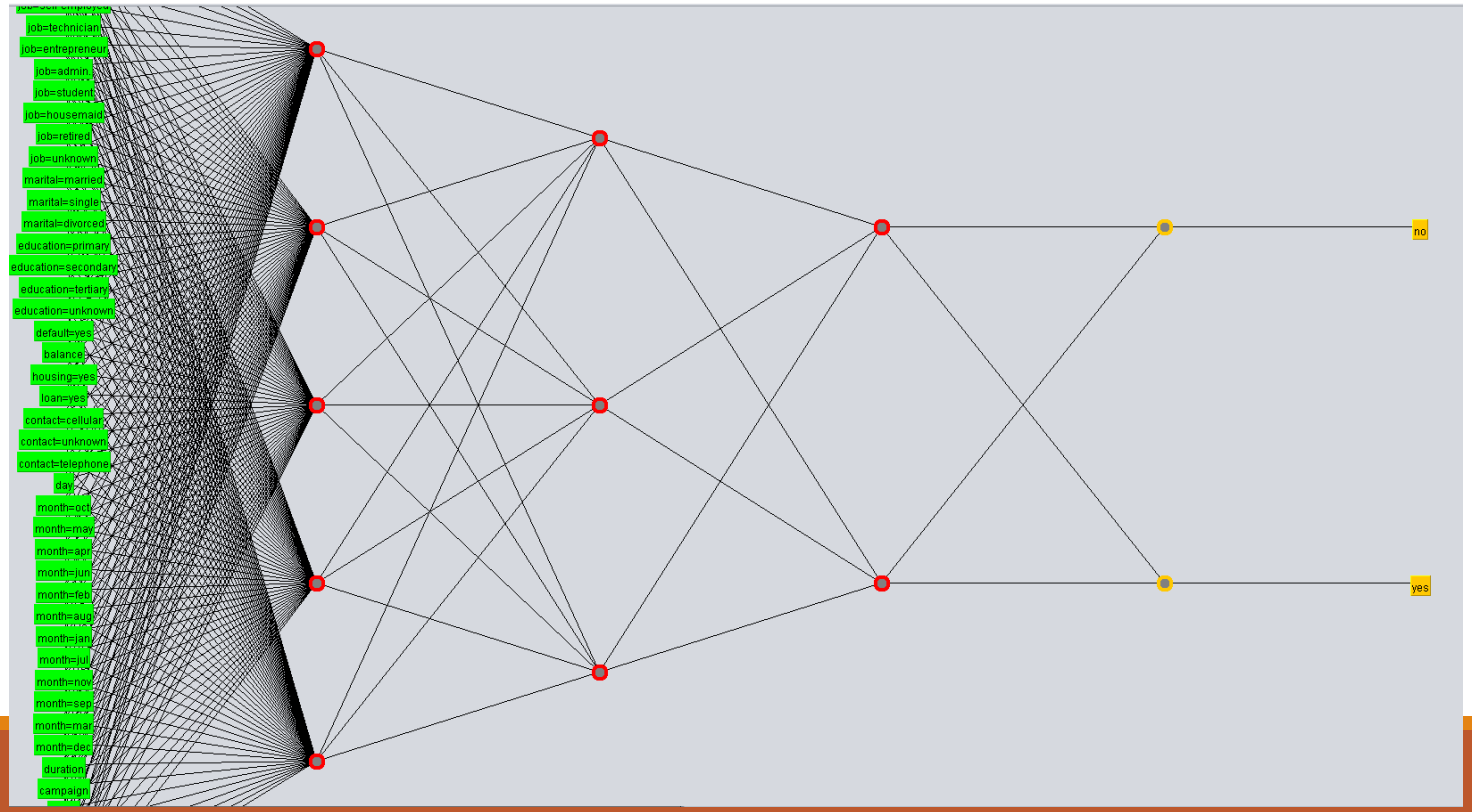
Let's change the hiddenLayers to 5,3,2

Click OK



Building Neural Network using GUI

After clicking Start, you can see now your network has 3 hidden layers where 5 nodes in layer 1, 3 nodes in layer 2 and 2 nodes in layer 3.



Remarks on Neural Network

The performance of the neural network can be easily affected by the setup of the **hyperparameters**. The hyperparameters include the number of hidden layers, number of nodes, learning rate, momentum, batch size, etc. To achieve a better performing neural network always requires tons of hyperparameters tuning. You can play with different hyperparameters setting and investigate which combination can achieve a better result.

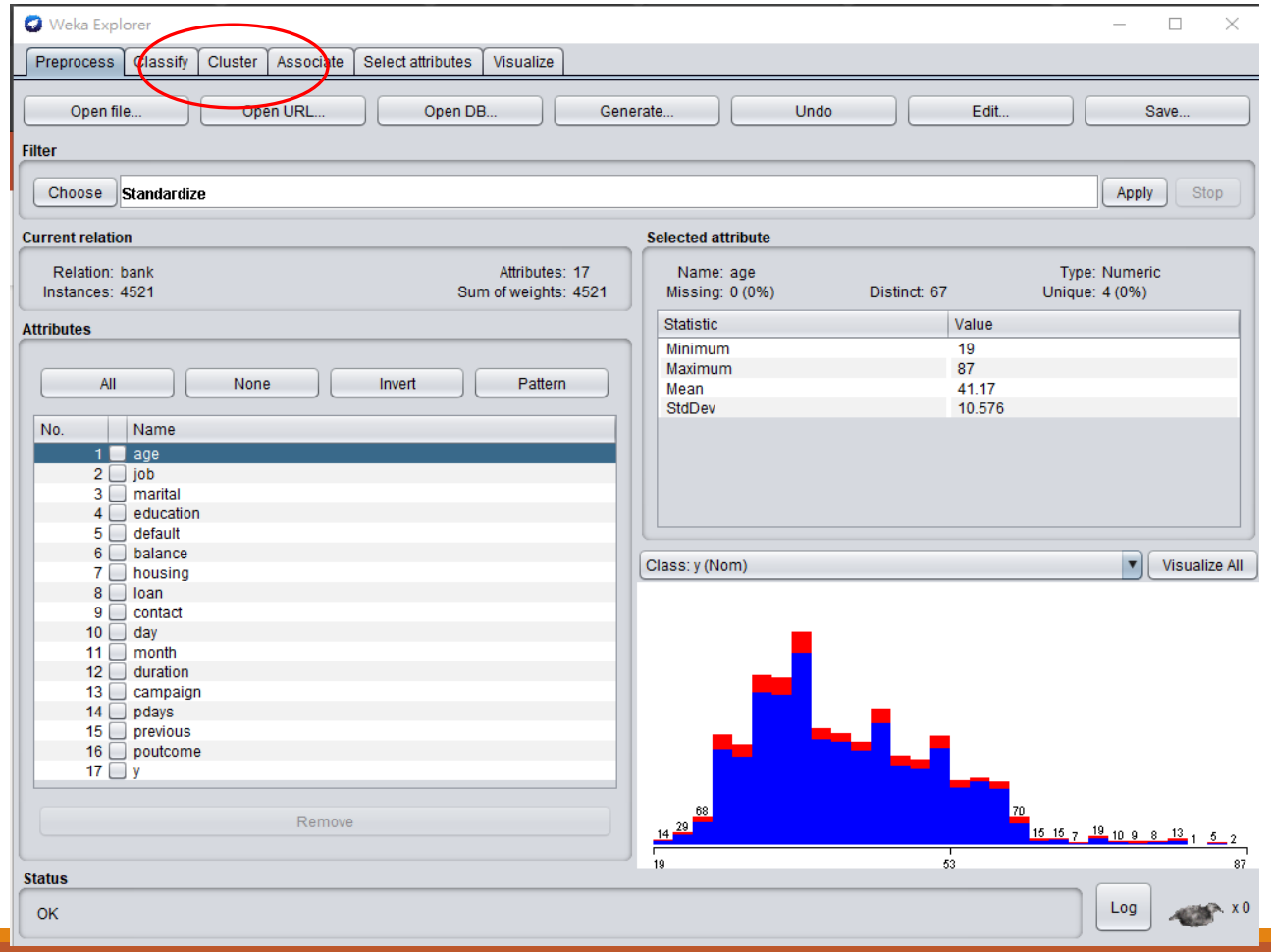
Clustering Algorithms with Weka

A clustering algorithm finds groups of similar instances in the entire dataset. WEKA supports several clustering algorithms such as EM, FilteredClusterer, HierarchicalClusterer, SimpleKMeans and so on. You should understand these algorithms completely to fully exploit the WEKA capabilities.

As in the case of classification, WEKA allows you to visualize the detected clusters graphically.

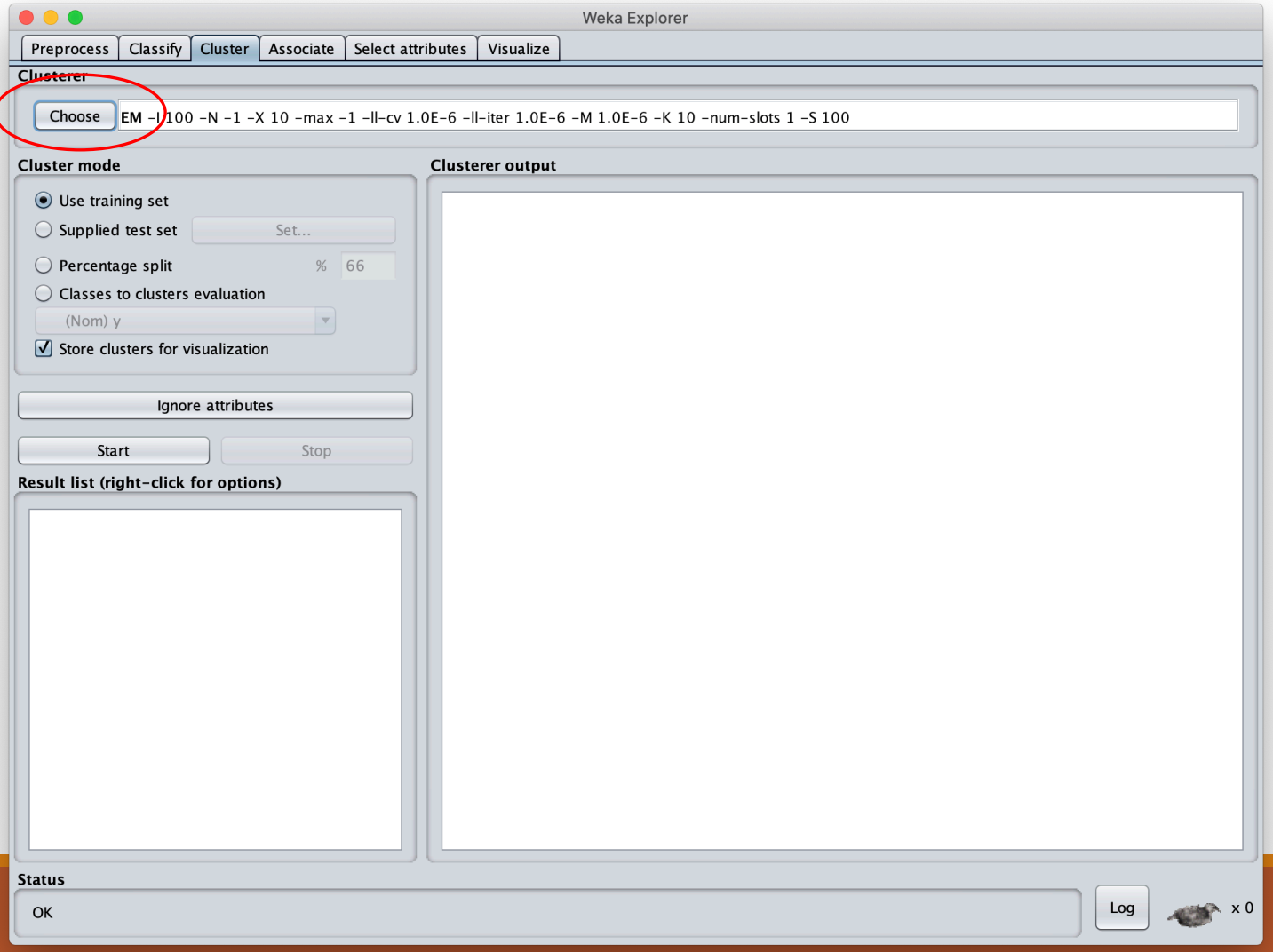
Clustering Algorithms with Weka

After loading data into Explorer, click Cluster



Clustering Algorithms with Weka

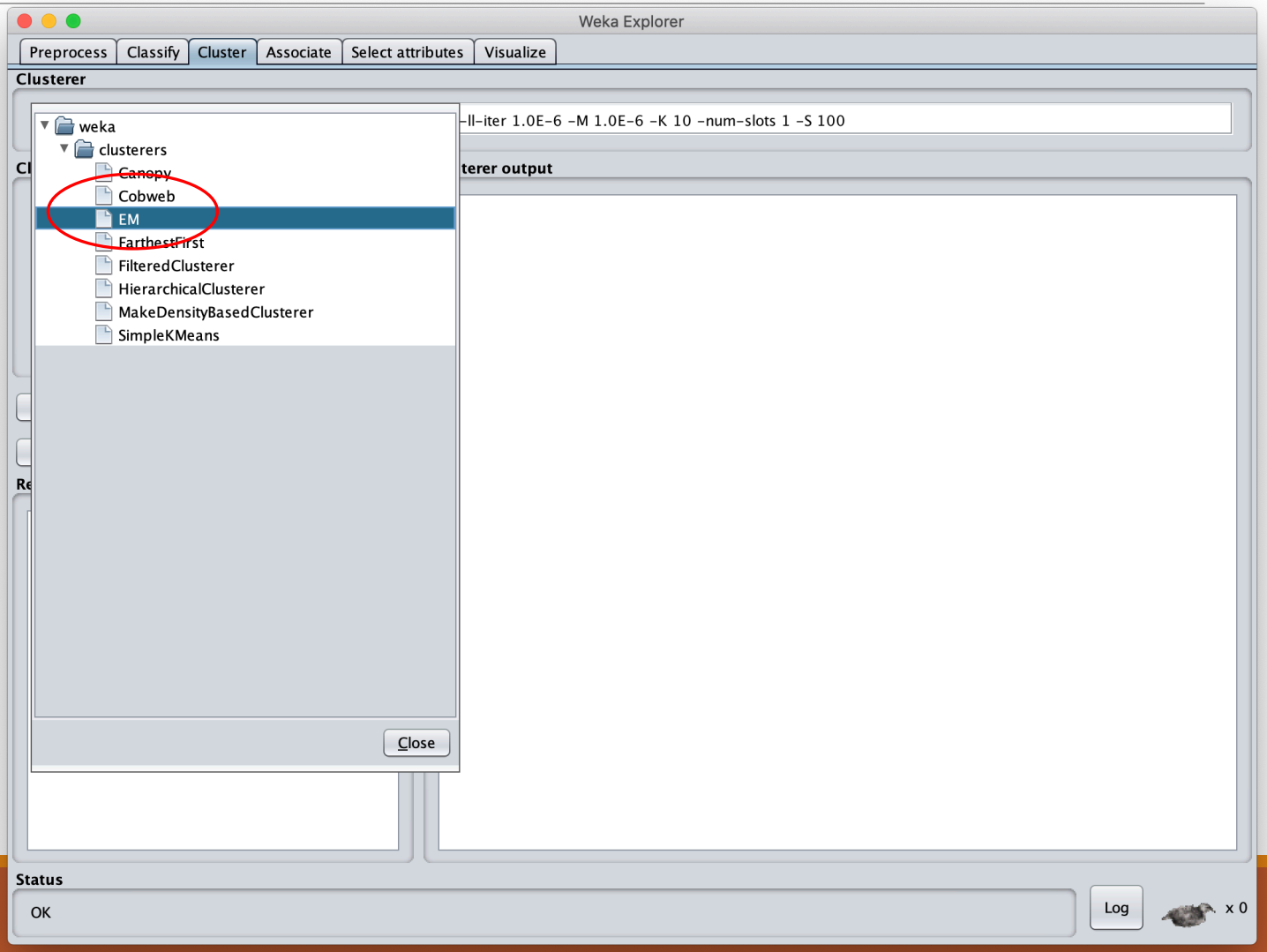
Click Choose



Clustering Algorithms with Weka

Under clusterers

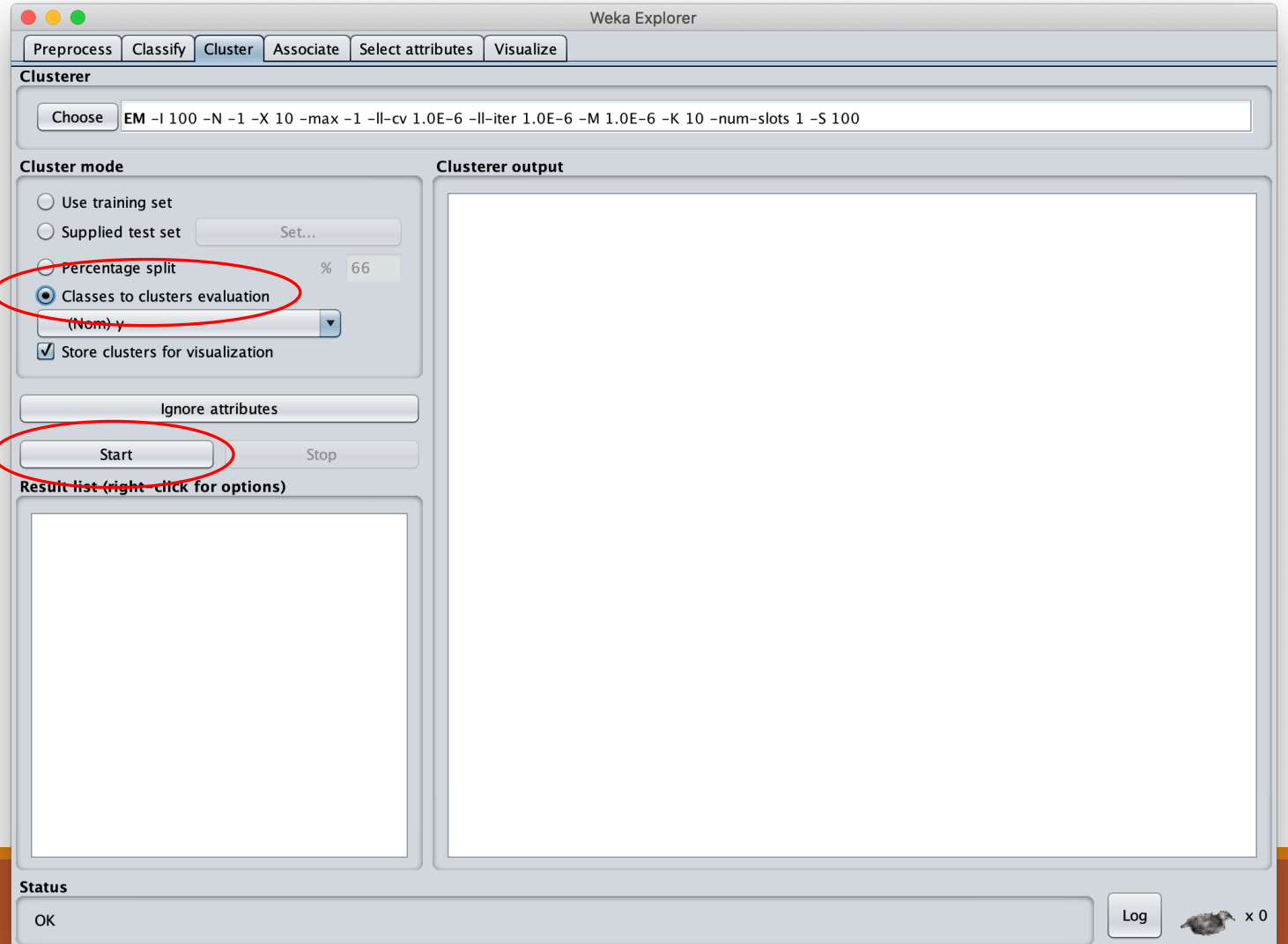
choose EM



Clustering Algorithms with Weka

In the **Cluster mode** sub window, select the **Classes to clusters evaluation** option.

Click on the **Start** button to process the data. After a while, the results will be presented on the screen.

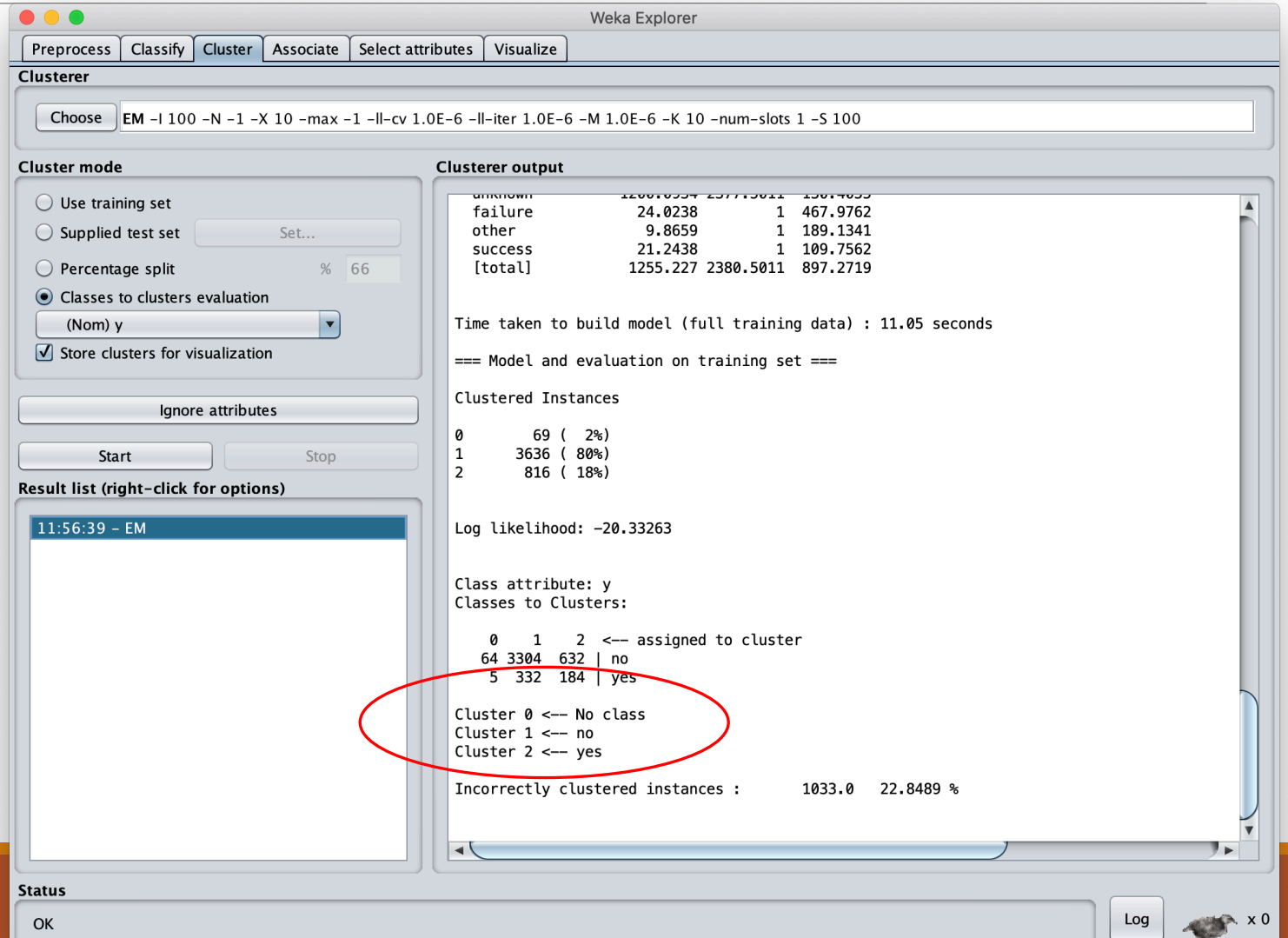


Clustering Algorithms with Weka

From the output screen, you can observe that –

There are 3 clustered instances detected in the database.

The **Cluster 0** represents no class, **Cluster 1** represents no, **Cluster 2** represents yes.

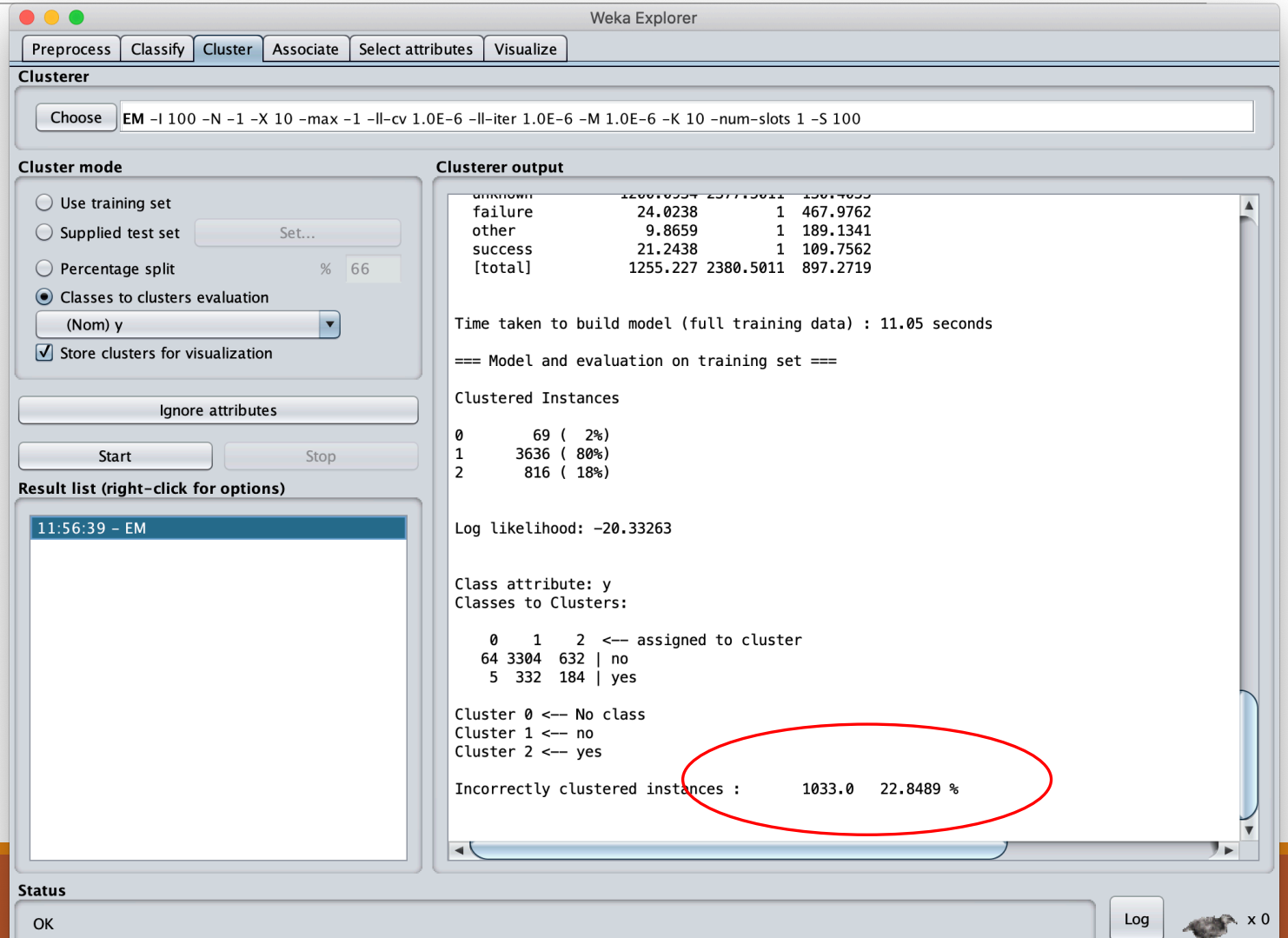


Clustering Algorithms with Weka

From the output screen, you can observe that –

Incorrectly clustered instances

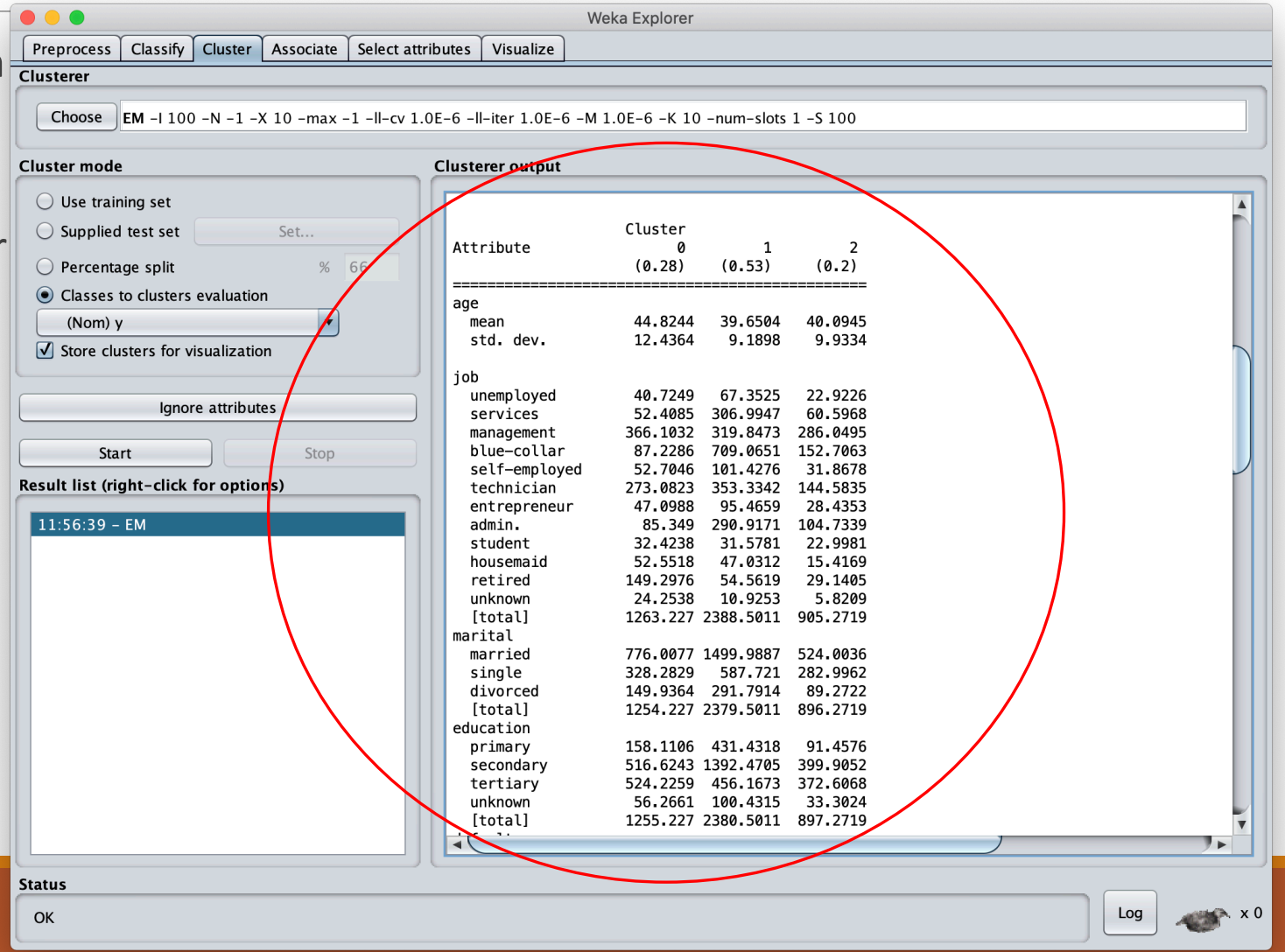
-> accuracy



Clustering Algorithms with Weka

From the output screen, you can observe that –

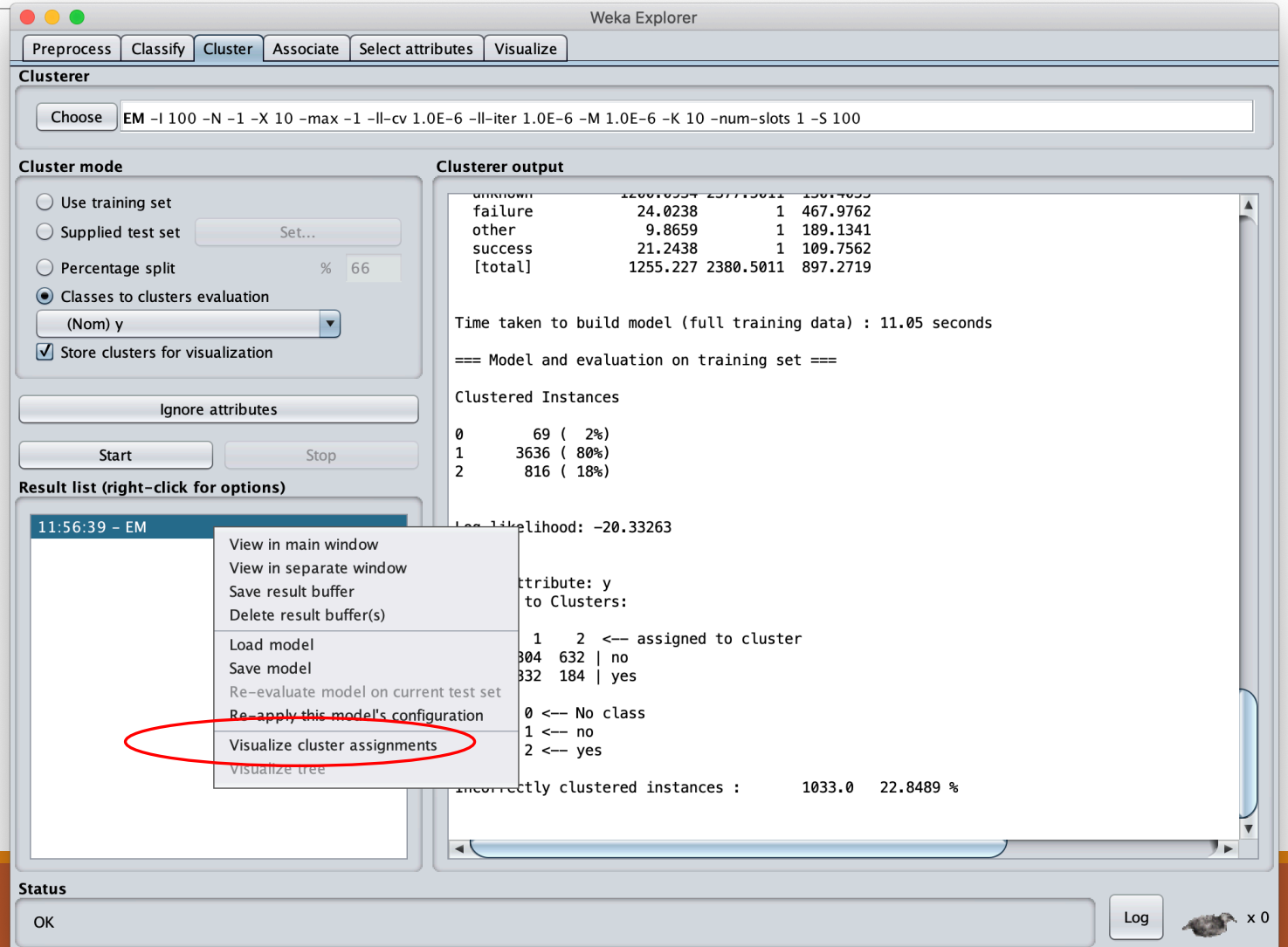
some statistics that gives the mean and standard deviation for each of the attributes in the various detected clusters.



Visualize Clusters with Weka

To visualize the clusters, right click on the **EM** result in the **Result list**. You will see the following options –

Select **Visualize cluster assignments**.



Visualize Clusters with Weka

You can play around by changing the X and Y axes to analyze the results. You may use jittering as in the case of classification to find out the concentration of correctly identified instances.