

**DATA MINING PROJECT 1**

**IMPLEMENTATION OF K-FOLD, KNN AND ANN ON STANDARD DATASETS**



October 31, 2015

CSE 5334

**VISHNU KIRAN GOLLAPUDI: 1001101334**  **NALAGATH SHANIA RASHEED:** **1001086536**

**WORK DISTRIBUTION:**

|  |  |
| --- | --- |
| NAMES | PART OF THE PROJECT WORKED ON |
| VISHNU KIRAN GOLLAPUDI: 1001101334 | **Implementation of the KNN AND K-FOLD using python.** |
| NALAGATH SHANIA RASHEED: 1001086536 | **Implementation of the ANN using the GUI Tool Weka, and making report.** |

**REPORT**

1. **INTRODUCTION:**

Nowadays there is huge amount of data being collected and stored in databases everywhere across the globe. The data is said to keep increasing year after year exponentially. It is not hard to find databases with Terabytes of data in enterprises and research facilities. That is over 1,099,511,627,776 bytes of data. There is a huge amount of private and invaluable knowledge “hidden” in such databases; and without the use of some kind of algorithms or procedures it is highly impossible to mine them. Throughout the years a large number of algorithms and hypothesis has been suggested and implemented for knowledge discovery from these extensive datasets. There exists several methodologies to extract what is called as the nuggets of knowledge, such as: classification, association rule, clustering, etc. In this project we will be dealing with one of the approach out of this that is classification technique.

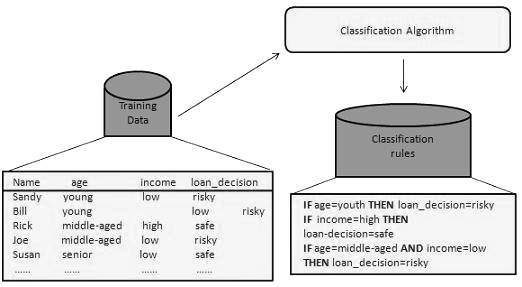
1. **CLASSIFICATION TECHNIQUE:**

Classification is a classic data mining technique based on machine learning. Basically classification is used to classify each item in a set of data into one of predefined set of classes or groups. It uses mathematical techniques such as decision trees, linear programming, neural network, k-nearest neighbor and statistics. In this technique we develop the software that can learn how to classify the data items into groups. For example, we can apply classification in application that “given all records of employees who left the company, predict who will probably leave the company in a future period.” In this case, we divide the records of employees into two groups that named “leave” and “stay”. And then we can ask our data mining software to classify the employees into separate groups. We have tried to do the same in this project as well.

* 1. **GENERAL WORKING OF A CLASSIFICATION TECHNIQUE:**

Classification consists of predicting a certain outcome based on a given input. It consists of two sets of data one is the training set containing a set of attributes and the respective outcome, which is usually called the goal or prediction attribute. It tries to map the relationships between the attributes that would make it possible to predict the outcomes. Next the algorithm is given a data set not seen before, called prediction set, which contains the same set of attributes, except for the prediction attribute – not yet known. The algorithm analyses the input and produces a prediction. The prediction accuracy defines how “good” the algorithm is.

Then, the classifier is built from the training set made up of database tuples and their associated class labels.

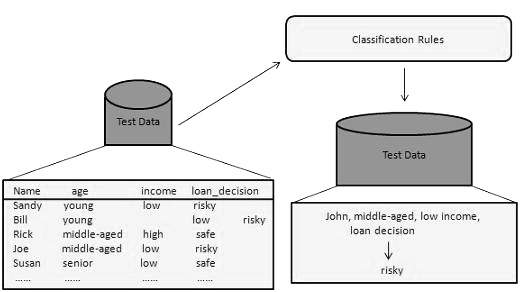


**Fig 1: the classification on a standard set**

* + 1. **Using Classifier for Classification:**

In this step we make use of the classifier built by training the data set. Here the test data is used to analyze how perfect the rules are by estimating the accuracy of these classification rules. The classification rules is applied to the new data

In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



**Fig2: the application of the classifier on a standard dataset**

1. **WHAT IS THE CLASSIFICATION PROBLEM WE ARE SOLVING?**

We are basically using 3 datasets and trying to classify them using two of the classification techniques that is:

* K-Nearest Neighbor
* Artificial neural network
  1. **K-Nearest Neighbor:**

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition already in the beginning of 1970’s as a non-parametric technique.

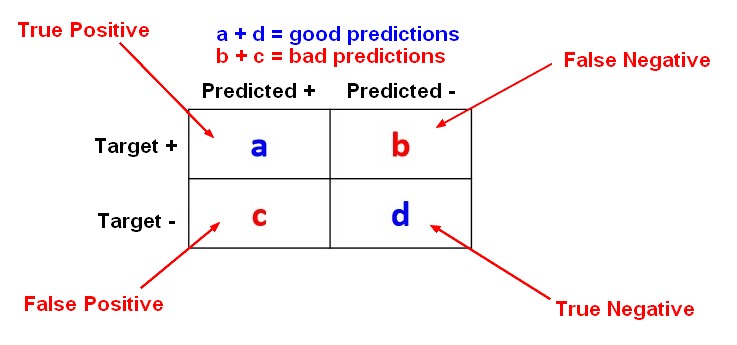
**Algorithm:**

Any data is classified by a majority vote of its neighbours, with the data point being assigned the class most common amongst its K-nearest neighbors measured by a distance function. If k=1, then the data point is simply assigned to the class of its nearest neighbor.

In order to see which neighbors are closest we use certain distance functions as given below:

|  |
| --- |
|  |
|  |
|  |
| http://www.saedsayad.com/images/KNN_similarity.png |
|  |
| There are exceptions while using these three distance measures i.e. they are valid only for continuous variables. In case of categorical variables the Hamming Distance is used. |
| http://www.saedsayad.com/images/KNN_hamming.png  Choosing the most optimal value for K can best be done by first inspecting the data. In general, if we select a large K-Value it is more precise as it reduces the overall noise but still there is no assurance about that.  Another technique that can be used to retrospectively determine a good K value is cross validation technique or so called K-fold Cross validation that we have implemented in our project.     * 1. **Artificial Neural Networks:**   One type of network sees the nodes as ‘artificial neurons’. These are called artificial neural networks (ANNs). An artificial neuron is a computational model inspired in the natural neurons. Natural neurons receive signals through synapses located on the dendrites or membrane of the neuron. When the signals received are strong enough (surpass a certain threshold), the neuron is activated and emits a signal though the axon. This signal might be sent to another synapse, and might activate other neurons. |
| https://science.education.nih.gov/supplements/nih2/addiction/images/guide/fig2.2.gif  **Fig 3: Neurons in the human body**  The complexity of real neurons is highly abstracted when modelling artificial neurons. These basically consist of inputs (like synapses), which are multiplied by weights (strength of the respective signals), and then computed by a mathematical function which determines the activation of the neuron. Another function (which may be the identity) computes the output of the artificial neuron (sometimes in dependence of a certain threshold). ANNs combine artificial neurons in order to process information.  **INPUT OUTPUT**  **(WEIGHTS)**  ACTIVATION FUNCTION   * + 1. **Multilayer Perceptron:**   The Multilayer Perceptron or as called the MLP many other neural networks basically make use of an algorithm called as the **backpropagation.**  A multilayer perceptron (MLP) is a finite acyclic graph. The nodes are neurons with logistic activation. With this backpropagation technique, the input data is repeatedly presented to neural network. The output of the neural network is compared to the desired output and the error is computed. This error is actually then feedback to the neural network and used to adjust the weights such that the error decreases with each iteration and the neural model gets more accurate in generating the desired output. This process is what we call as **“Training”**.   * The three layers are as follows:   **Input Layer** — A vector of predicatble variable values (x1...xp) is presented to the input layer. The input layer standardizes these values so that the range of each variable is from -1 to 1. The input layer distributes the values to each of the neurons in the hidden layer.  **Hidden Layer** — Evualating the neuron in the hidden ayer, the value from each of the neurins in the input layer is multiplied by weight and the resulting weighted values are added together producing a combined value.  **Output Layer** — Evaluating a neuron in the output layer, the value from each hidden neuron is multiplied by a weight and the resulting weighted values are added together on which we apply the transfer function. This is called the outputs of the network.  https://www.dtreg.com/uploaded/pageimg/MLFNwithWeights.jpg  **Fig 4: Multilayer Perceptron**   1. **RESULT EVALUATION:**   The result that is produced i.e. the classification results that are generated has been evaluated using the technique called as the K-Fold Cross Validation technique.   * **K-Fold Cross Validation:**   K-fold Cross Validation is a common technique for estimating the performance of a classifier. Given a set of m training examples, a single run of k-fold cross validation proceeds as follows:   1. Arrange the training examples in a random order. 2. Divide the training examples into k folds. 3. For i = 1, . . . , k:  * Train the classifier using all the examples that do not belong to Fold i. * Test the classifier on all the examples in Fold i. * Compute ni , the number of examples in Fold i that were wrongly classified.  1. Return the following estimate to the classifier error:   To obtain an accurate estimate to the accuracy of a classifier, k-fold cross validation is run several times, each with a different random arrangement in Step 1. Let E1, . . . , Et be the accuracy estimates obtained in t runs.  **Define:**  The estimate for the algorithm performance is an error of e with standard-deviation of σ.   * **Accuracy:** |

Accuracy is one of the quantitative measures which gives us the percentrage as to how accurate is the predictive model. There are various measures in accuracy, but all of them are directly dependent on the data that is being used. In real time environment there is a chance that the values may be missing or may contain null values or it may have been changed or transformed according to different processes being used on them. For instance, a sales predictive model for any large store based on its history of sales can be strongly correlated and accurate but the data in that can be wrong. Therefore, measurements of accuracy is balanced by assessments of reliability.



**Fig 5: Confusion Matrix**

Hence the accuracy is being calculated using the following formulae:

1. **DATASETS USED:**

We have made use of total 3 data sets:

* 1. **ATT Dataset:**

This is the dataset that was provided to us by preprocessing a huge data set of 40 AT&T users. It is a numeric dataset which is provided with the corresponding labels.

**Dimensions:** 645 x 401

* 1. **IRIS Dataset:**

This is the standard and one of the most popular datasets that is used worldwide to perform various data mining methodologies. It basically consists of 2 types of attributes that is the numeric and the nominal.

**Dimensions:** 5 x 151

**Attributes:** Sepal.Length, Sepal.width, Petal.length, Petal.width & Species.

**Classes:** It can be classified in the following classes: setosa, versicolor, virginica.

**Referenced:** *https://code.google.com/p/dataminingproject/source/browse/DataMiningApp/datasets/Iris/iris.csv?r=44*

* 1. **Blood Transfusion Dataset:**

To demonstrate the RFMTC marketing model (a modified version of RFM), this study adopted the donor database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. The center passes their blood transfusion service bus to one university in Hsin-Chu City to gather blood donated about every three months. To build a FRMTC model, we selected 748 donors at random from the donor database. These 748 donor data, each one included R (Recency - months since last donation), F (Frequency - total number of donation), M (Monetary - total blood donated in c.c.), T (Time - months since first donation), and a binary variable representing whether he/she donated blood in March 2007 (1 stand for donating blood; 0 stands for not donating blood).

**Dimensions:** 5 x 749

**Attributes:** Recency (months), Frequency (times), Monetary (c.c. blood), Time (months), whether he/she donated blood in March 2007

**Classes:** 0, 1.

**Referenced:**

*http://archive.ics.uci.edu/ml/datasets/Blood+Transfusion+Service+Center*

1. **IMPLEMENTATION:**

The KNN with the K-FOLD evaluation has been implemented using the language python. The following are resultant of all the 3 datasets:

* 1. ATT Dataset for K=5 and 10:

This accuracy has been achieved by shuffling of the dataset.

C:\Users\shania\Downloads\images 5.18.20 PM\att-5.tiff

C:\Users\shania\Downloads\images 5.18.20 PM\att-10.tiff

* 1. IRIS Dataset for k=5 and 10:

C:\Users\shania\Downloads\images 5.18.20 PM\iris-5.tiff

C:\Users\shania\Downloads\images 5.18.20 PM\iris-10.tiff

* 1. Transfusion Dataset for K=5 and 10:

C:\Users\shania\Downloads\images 5.18.20 PM\transfusion-5.tiff

C:\Users\shania\Downloads\images 5.18.20 PM\transfusion-10.tiff

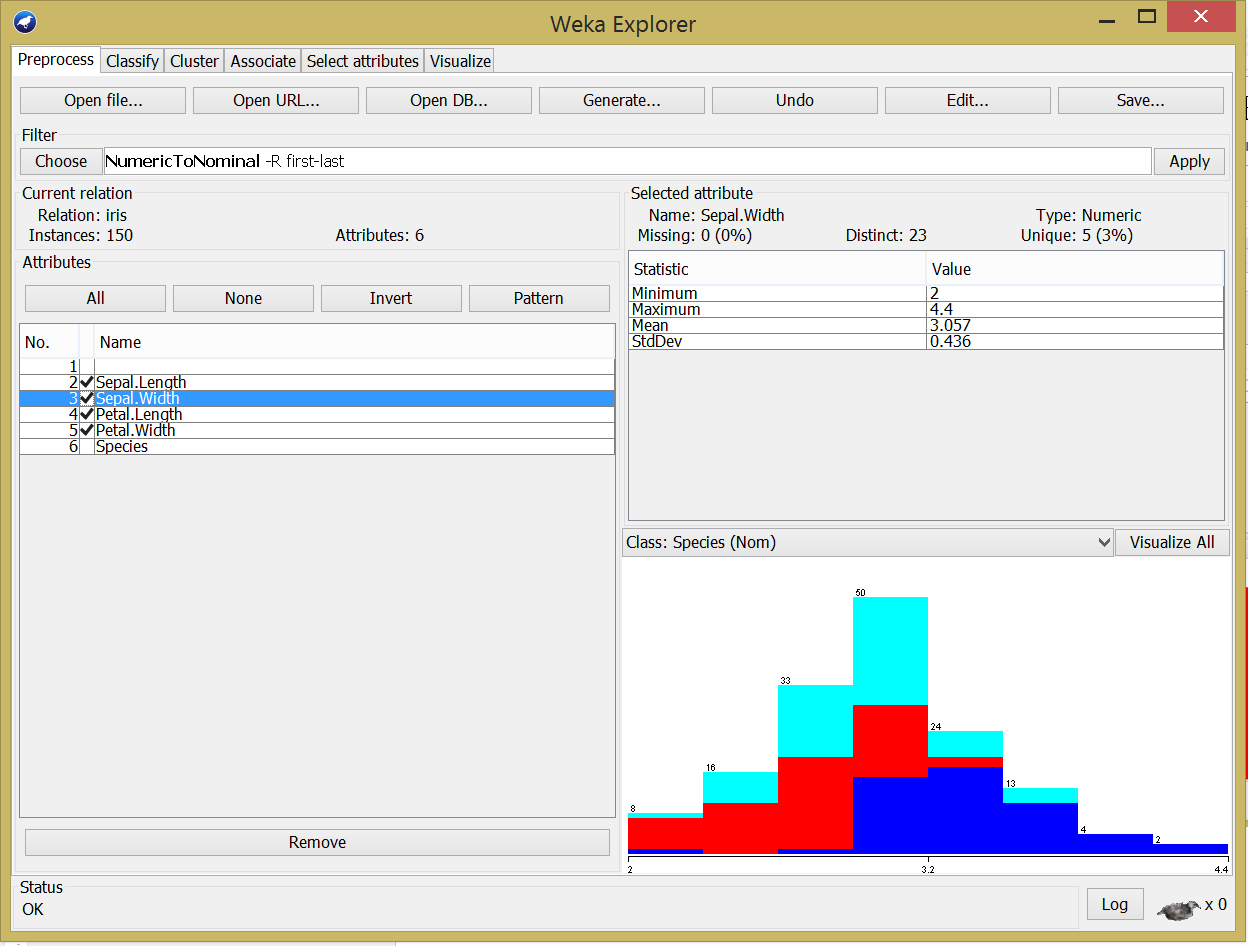
The ANN has been implemented in the tool Weka 3-6.

* **WEKA: Weka** is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from your own Java code. **Weka** contains **tools** for data pre-processing, classification, regression, clustering, association rules, and visualization.

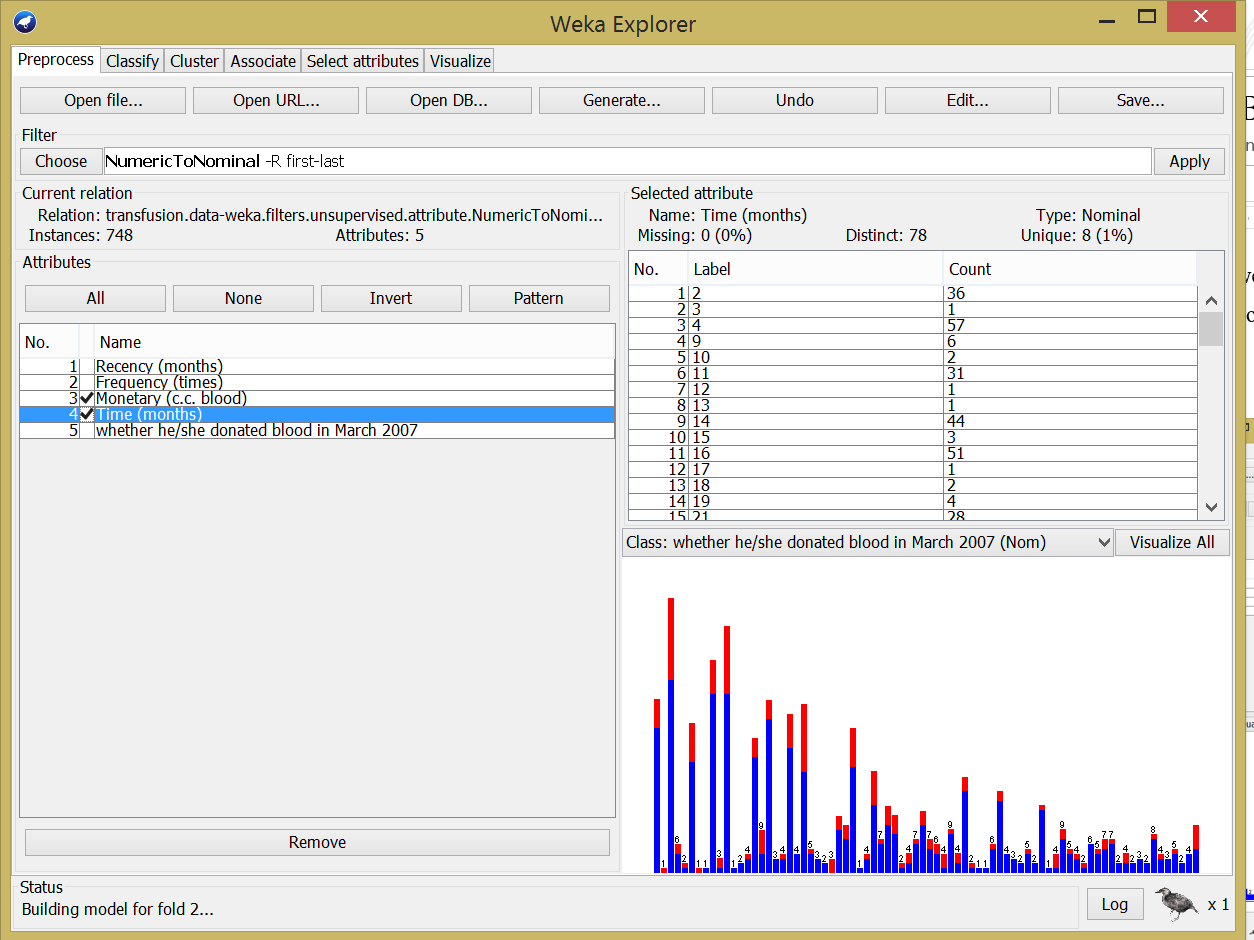
**STEPS:**

In order to run the Multilayer Perceptron in Weka. The first step we need to do is presprocess the data in it by changing all the datasets from numeric to nominal if there is any. We need to load the csv file into weka first.

* **IRIS DATASET:**



* **ATT DATASET:**
* **TRANSFUSION DATASET:**

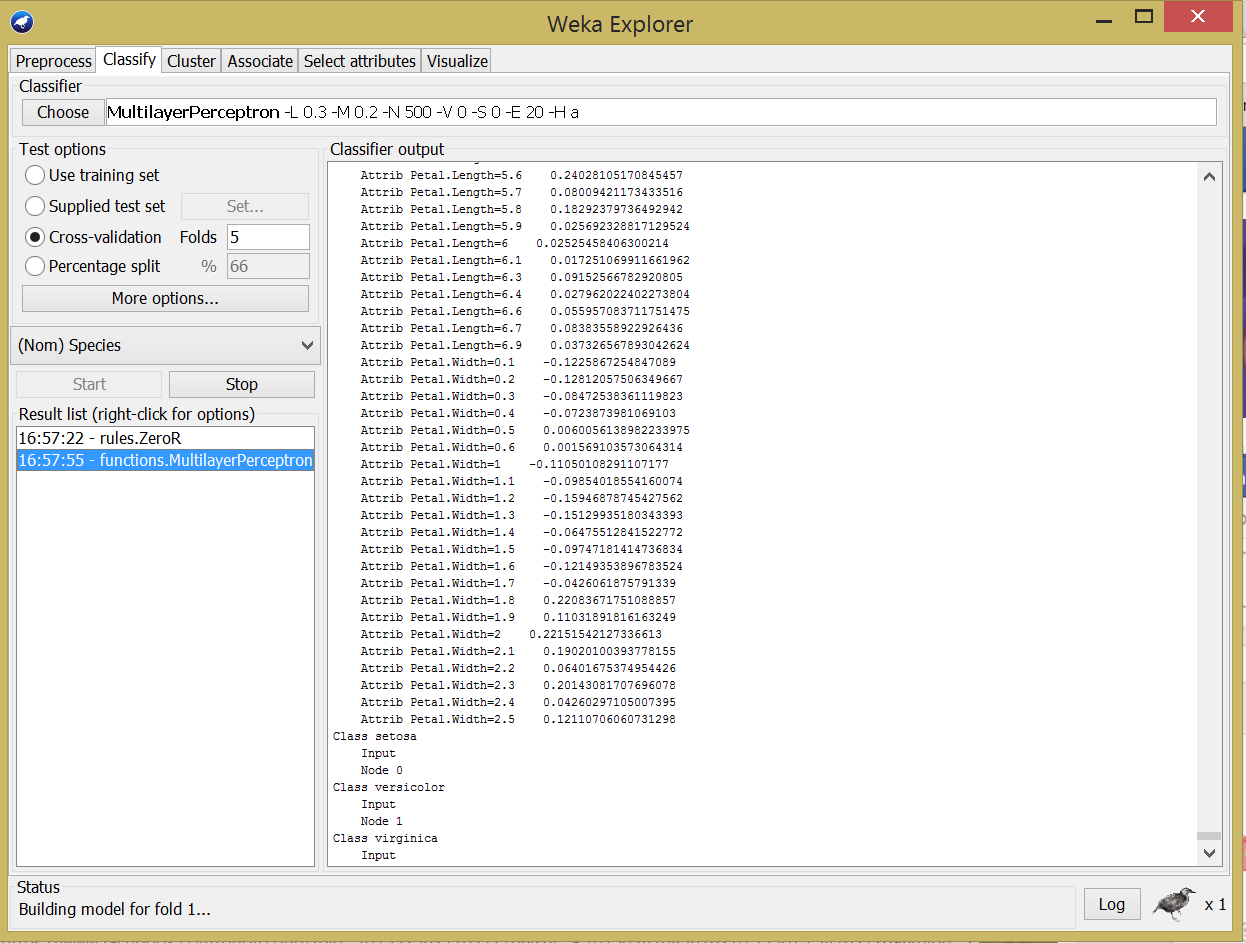


As we can see in the above image we select the, attributes that are numeric and select the unsupervised learning technique of numeric to nominal and then click on apply. Then this file is saved with the extension “.arff” extension and all the functions are performed on this transformed dataset.

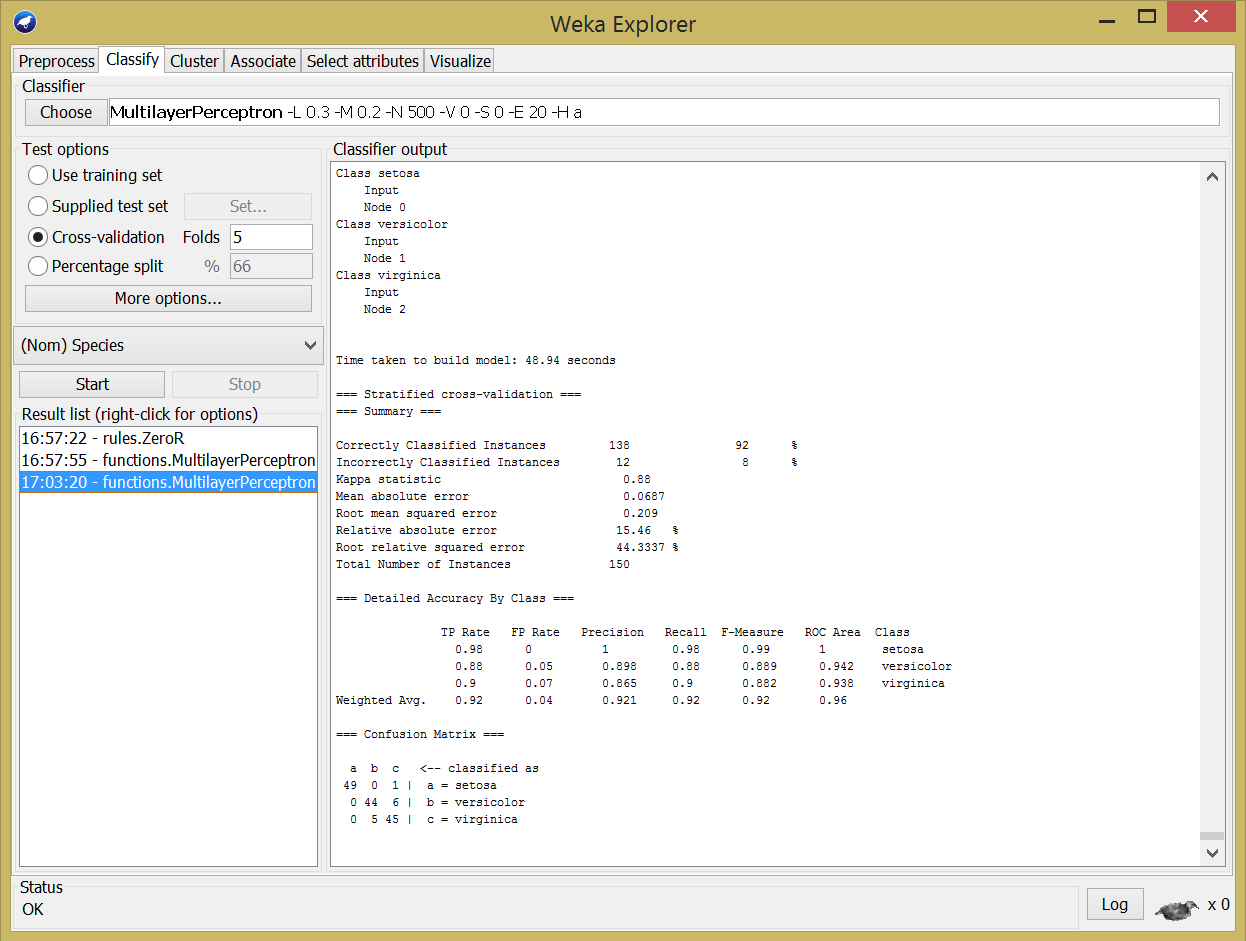
Next, we need to select the “classify” tab and select the functions from the classifier as “Multi Layer perceptron” an select the various K-fold options as 5, or 10 or whatever value you need and select the class label from the tab and click on start. We can change the number of hidden nodes as well, but here we are suing the average of the number of input and output nodes.

* **IRIS DATASET:**

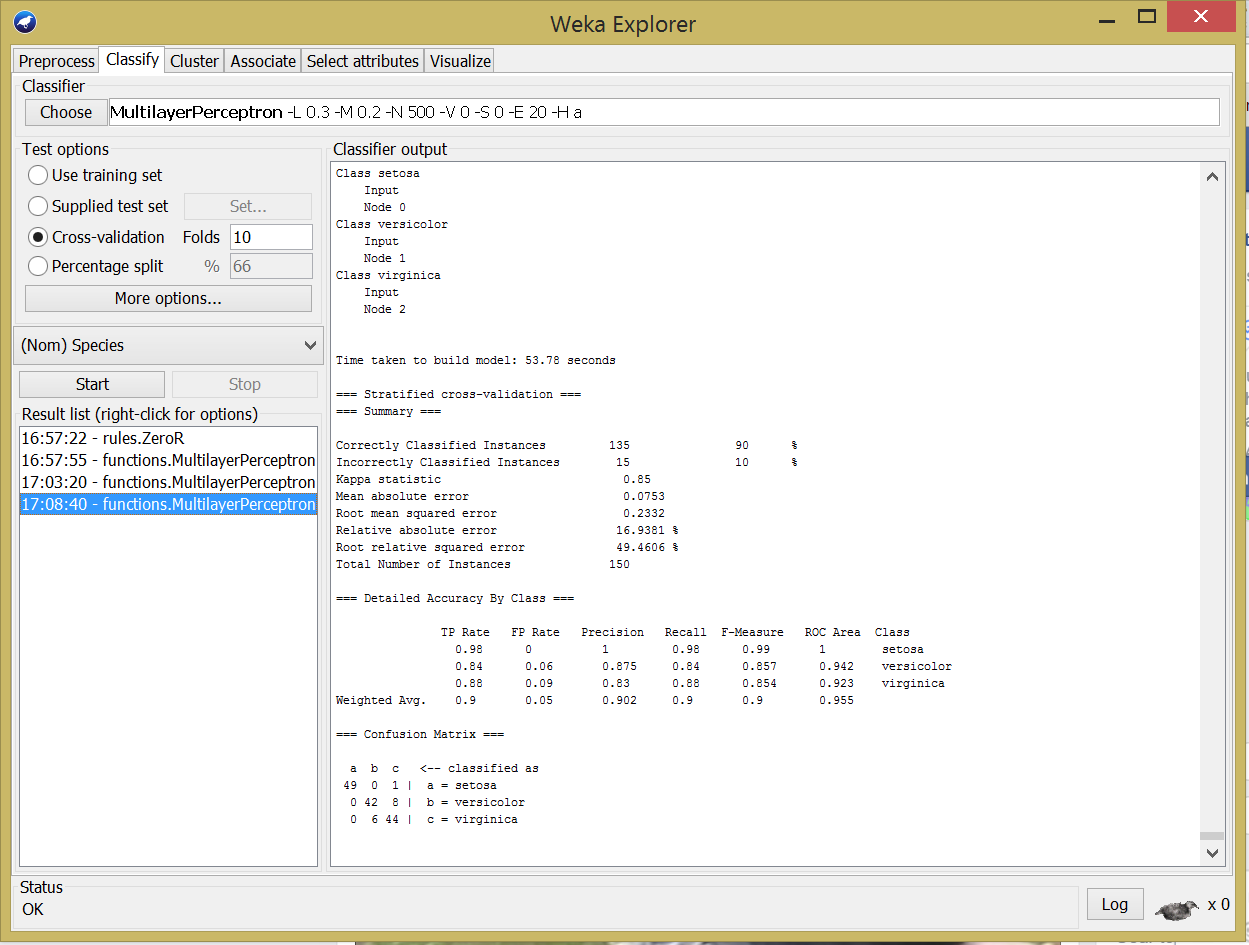
It will build the model for each fold, for instance for **fold =1** is as follows:



Accuracy for Fold=5

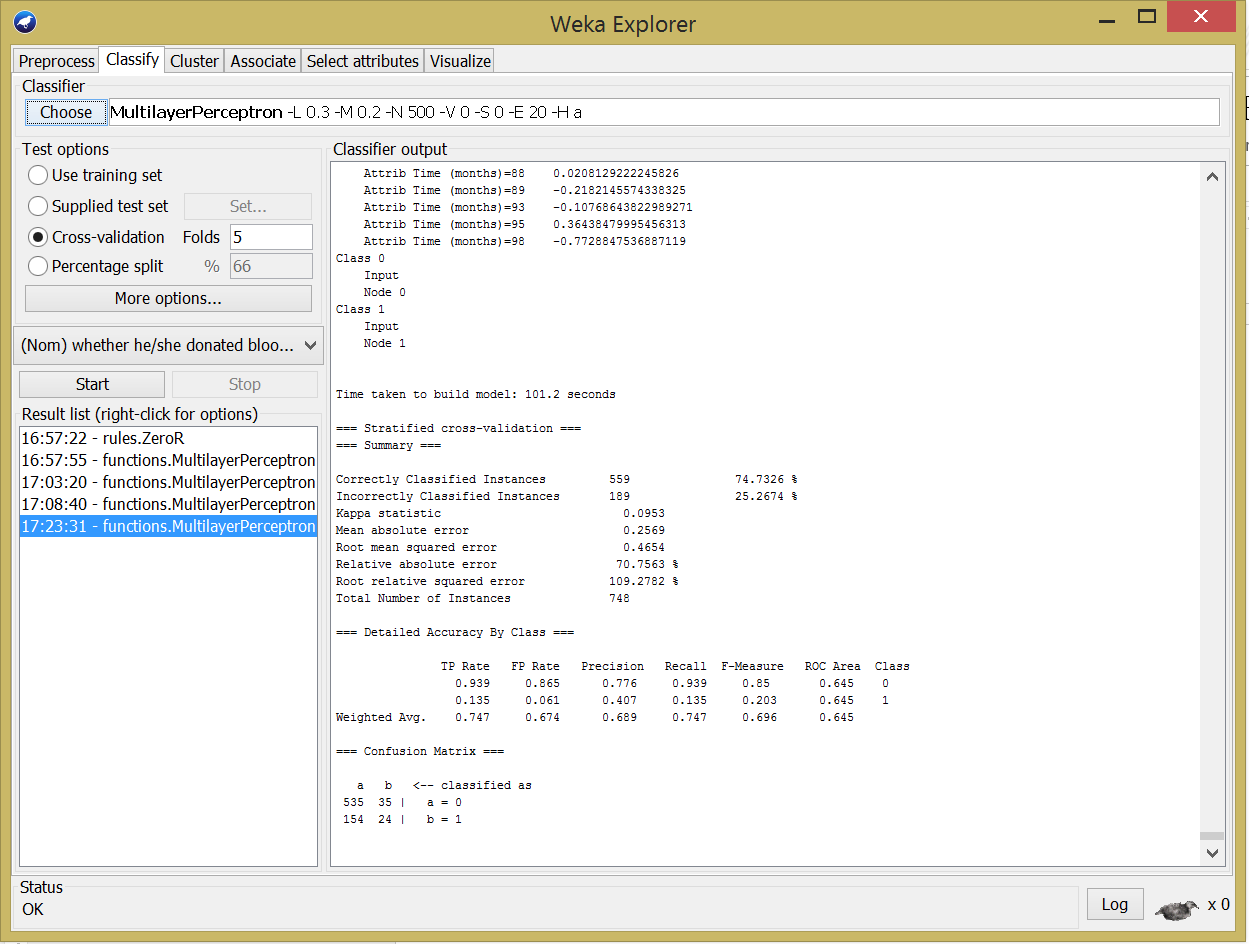


Accuracy for fold=10

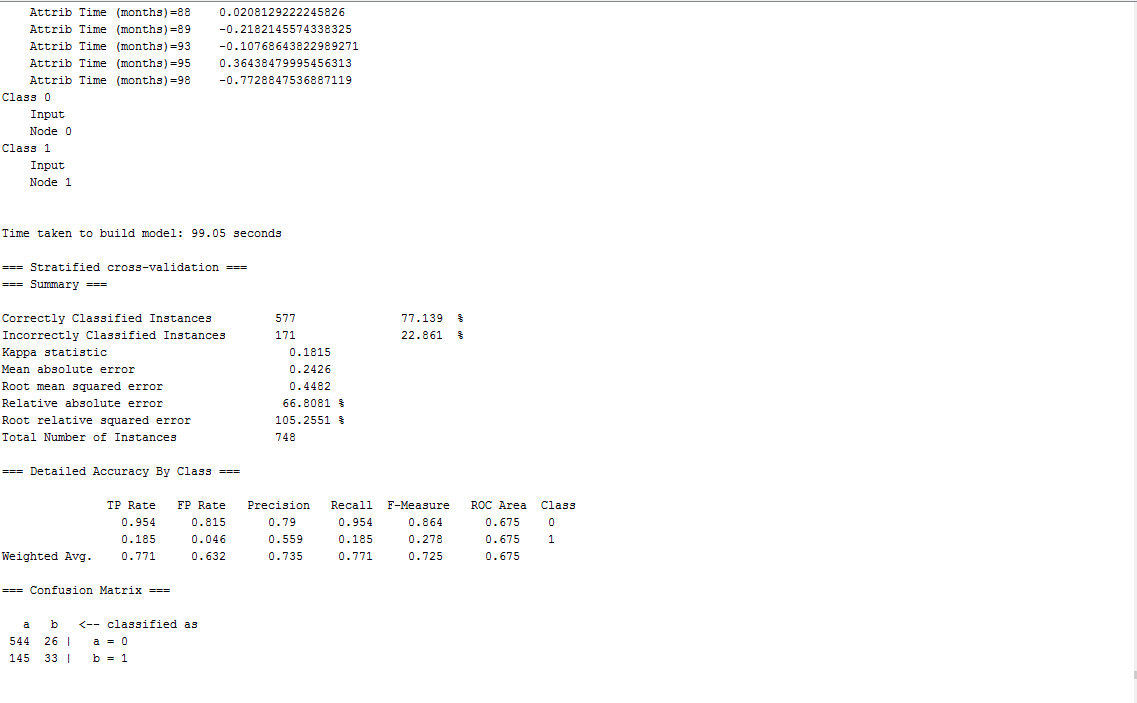


* **TRANSFUSION DATASET:**

Accuracy for Fold=5

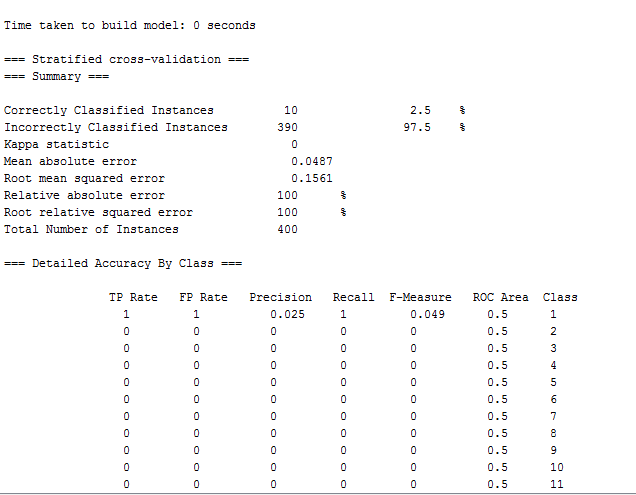


Accuracy for fold=10

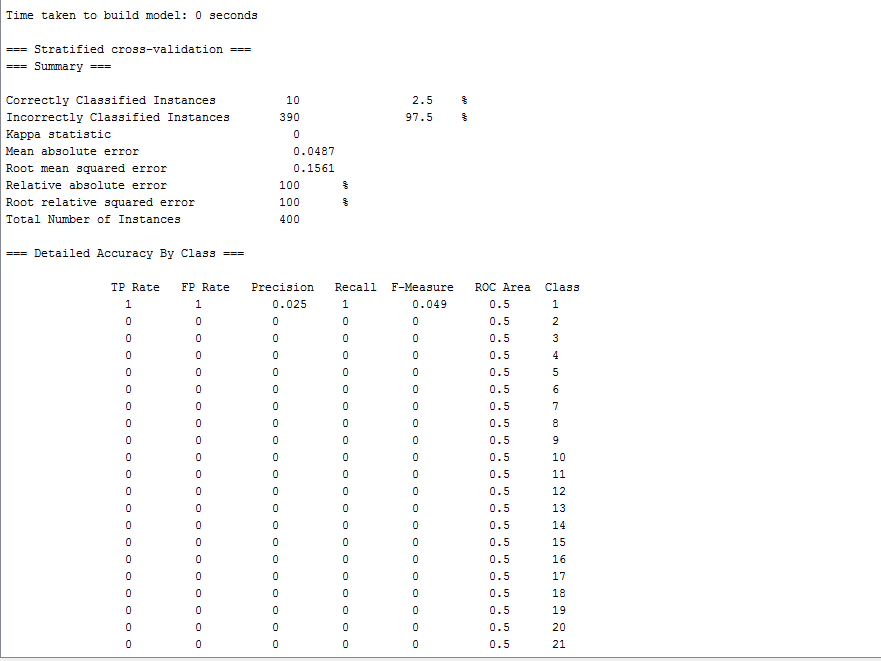


* **ATT DATASET:**

Accuracy for Fold=5



Accuracy for fold=10



**REFERENCES:**

1. <http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>
2. [https://datajobs.com/data-science-repo/Neural-Net-[Carlos-Gershenson].pdf](https://datajobs.com/data-science-repo/Neural-Net-%5bCarlos-Gershenson%5d.pdf)
3. <http://www.csie.ntu.edu.tw/~b92109/course/Machine%20Learning/Cross-Validation.pdf>
4. <http://scikit-learn.org/stable/modules/generated/sklearn.cross_validation.KFold.html>
5. <http://www.mu-sigma.com/analytics/thought_leadership/cafe-cerebral-neural-network.html>
6. <http://ml.informatik.uni-freiburg.de/_media/documents/teaching/ss09/ml/mlps.pdf>
7. <http://machinelearningmastery.com/tutorial-to-implement-k-nearest-neighbors-in-python-from-scratch/>
8. <https://msdn.microsoft.com/en-us/library/ms174493.aspx>
9. http://gerardnico.com/wiki/data\_mining/accuracy