NEC - Exercise 3

Unsupervised learning with SOM, PCA and k-means

Carlos García

February 2017

Contents

Exercise 3: Unsupervised learning with SOM, PCA and k-means

Objective

Compare the results from three different unsupervised learning techniques:

- Principal Component Analysis (PCA)
- K-means
- Self-Organizing Maps (SOM)

1 Main concepts

1.1 Iris flower data set

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Ronald Fisher in his 1936 paper The use of multiple measurements in taxonomic problems as an example of linear discriminant analysis. It is sometimes called Anderson's Iris data set because Edgar Anderson collected the data to quantify the morphologic variation of Iris flowers of three related species. Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus". The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other [1].

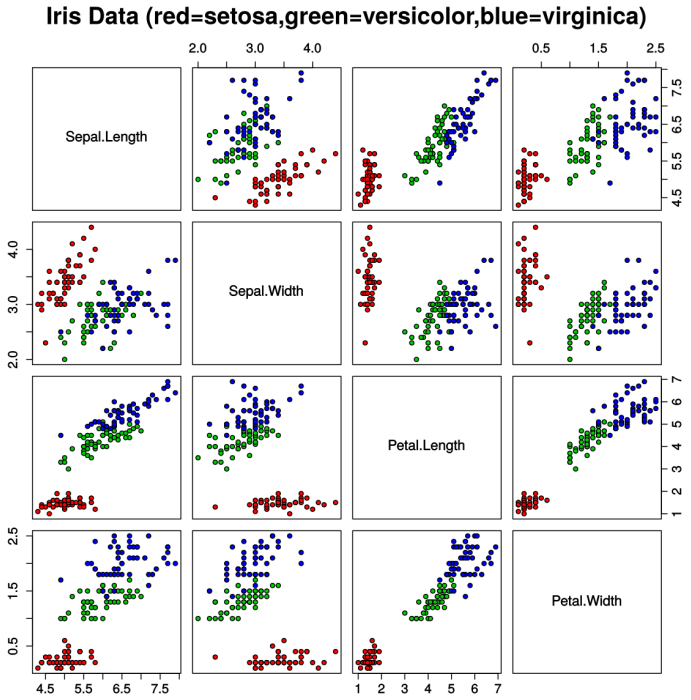


Figure 1: Scatterplot of the data set [1]

1.2 Principal Component Analysis (PCA)

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to the preceding components. The resulting vectors are an uncorrelated orthogonal basis set. PCA is sensitive to the relative scaling of the original variables [2].

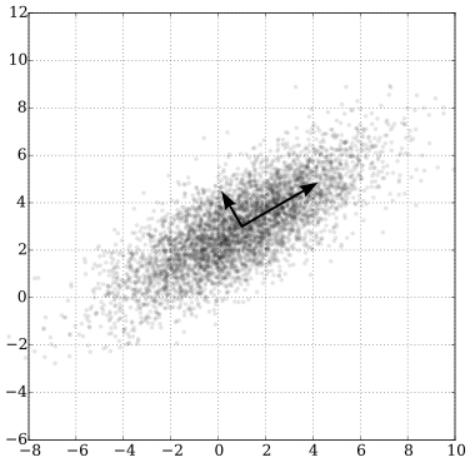


Figure 2: PCA of a multivariate Gaussian distribution [2]

1.3 K-means

k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells [3].

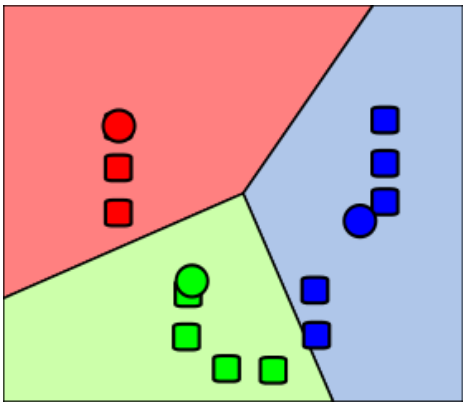


Figure 3: Kmeans example clustering [3]

1.4 Self-Organizing Maps (SOM)

A self-organizing map (SOM) or self-organizing feature map (SOFM) is a type of artificial neural network (ANN) that is trained using unsupervised learning to produce a low-dimensional (typically two-dimensional), discretized representation of the input space of the training samples, called a map, and is therefore a method to do dimensionality reduction. Self-organizing maps differ from other artificial neural networks as they apply competitive learning as opposed to error-correction learning (such as backpropagation with gradient descent), and in the sense that they use a neighborhood function to preserve the topological properties of the input space [4].

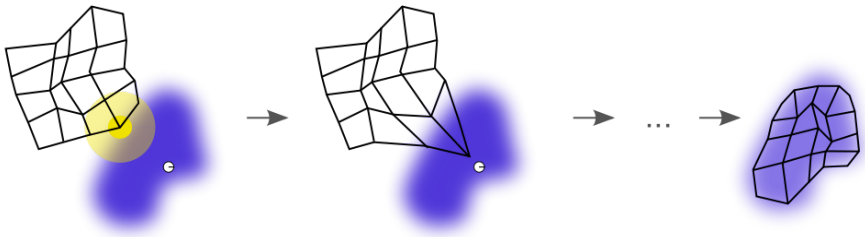


Figure 4: An illustration of the training of a self-organizing map. The blue blob is the distribution of the training data, and the small white disc is the current training datum drawn from that distribution [4]

2 Results obtained

As for the prior exercises, I did this clustering analysis using Weka. After some experiences working with unsupervised learning for another subjects (Computer Vision and Biometrics Identification), we have learned than for the cases of clustering and unsupervised learning is always better start the work by analyzing the data and try to find the features which are more distinguishable for the classes or those which have more variance.

First we go to Weka and import the data which was replicated into a csv file:

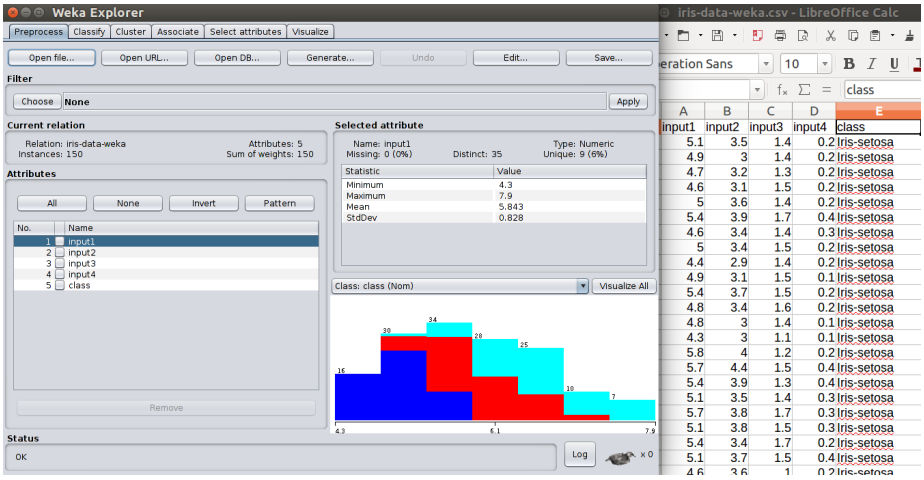


Figure 5: Openning data into Weka

In this window we can press “Visualize all” and this will show us the behavior for all the attributes:

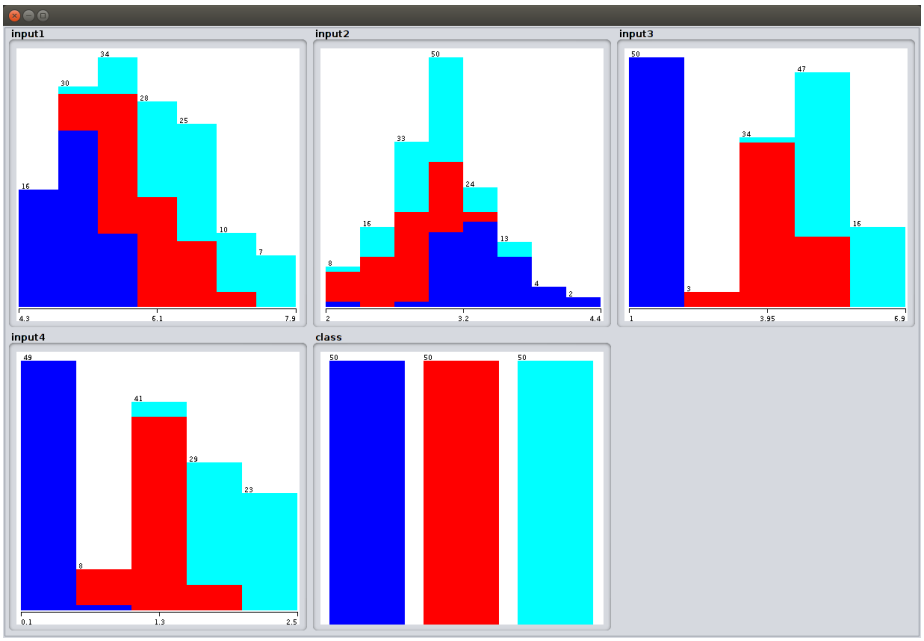


Figure 6: All data visualization for study

Watching this plots, we can have a better interpretation of the inputs, for example now we know that inputs 1 and 2 are more merged and maybe hard to analyze than inputs 3 and 4. Additionally in input 3 we see how every class has at least one portion of the values where it is alone, it means that for instance, if we are making a clostering and input 3 equals a value between 1 and 3 we already know that this item belongs to "blue" (iris-setosa) class. This kind of information is relevant before starting because gives us more confidence about final results.

Then we can also go to "Visualize" tab and check a plot representation of the attributes with respect to each others, this is also useful to identify the good relations between attributes when clustering.

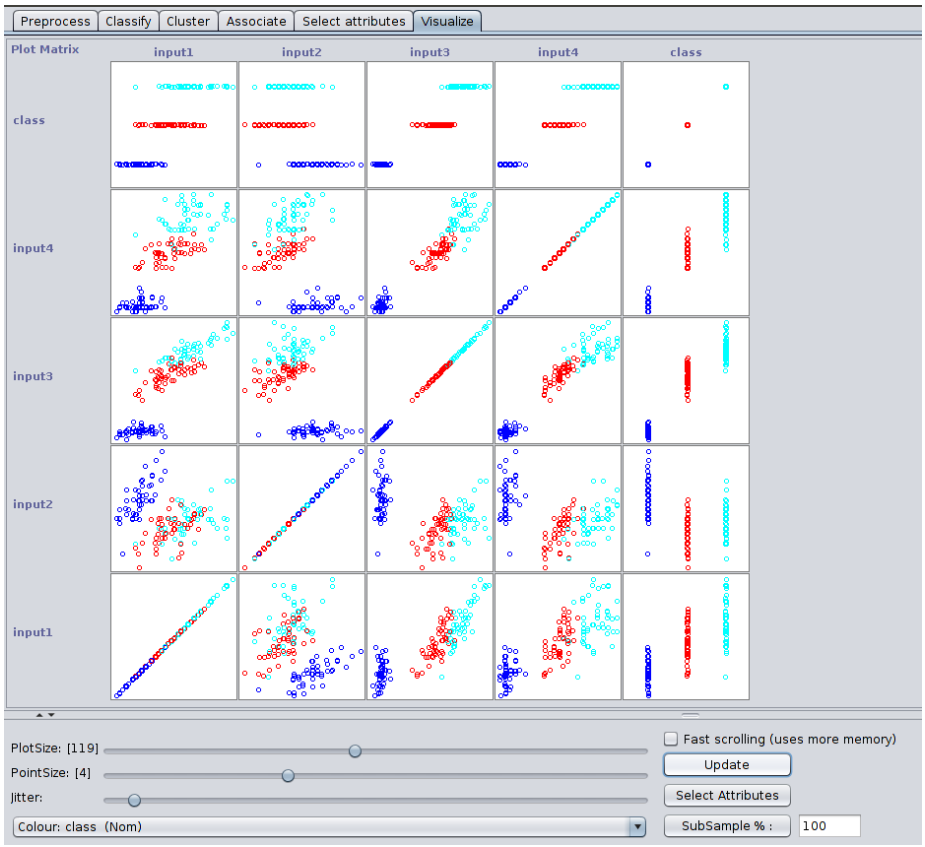


Figure 7: Matrix of relationships between all attributes

If we want to see a relationship in detail we simply click over the block containing the relationship we are interested in.

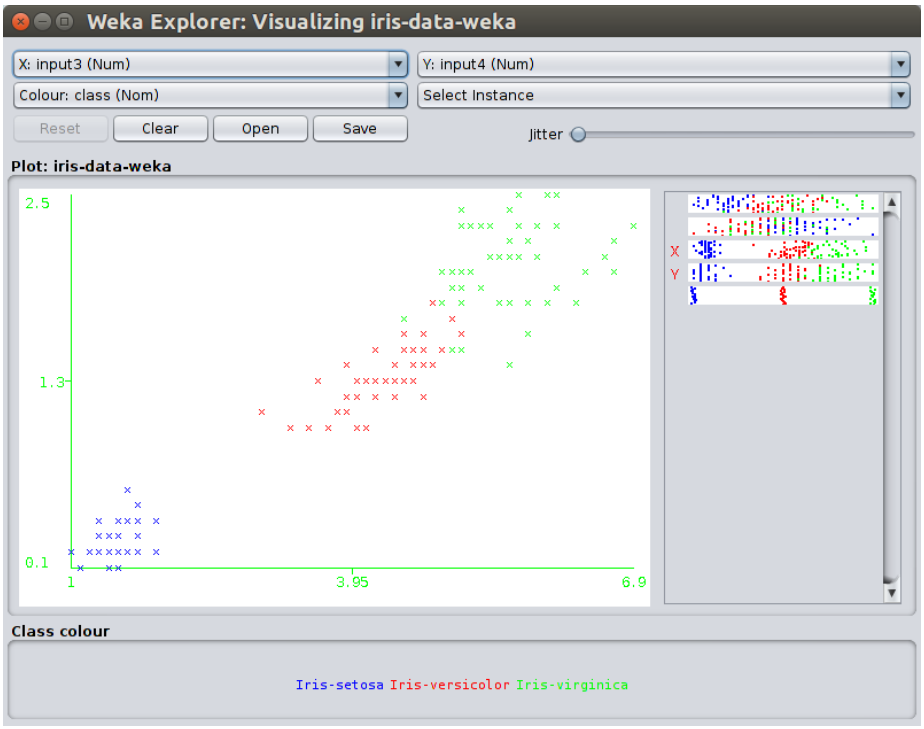


Figure 8: Detailed relationship between input 3 and 4

In this case for example the combination of inputs 3 and 4 gives us some good separable clusters.

Another popular way of analyzing input data using Weka is by using a decision tree, in this case we can try showing one by selecting the Tree/J48 classifier into Classify tab:

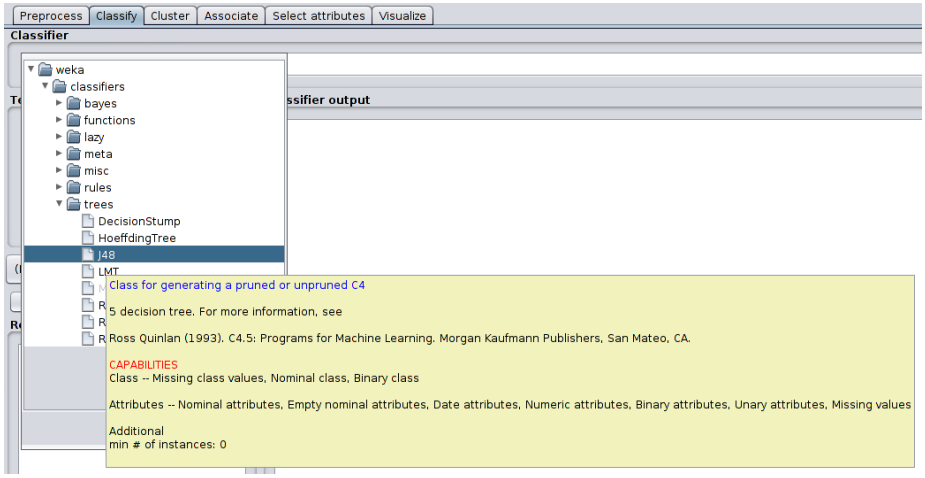


Figure 9: Selecting J48 tree classifier

This decision tree helps to know which decisions to make when we are classifying a dataset, if we press the start button it trains and returns some useful information like the summary, the accuracy, the confusion matrix and the best part is the graphical tree which can be seen by making right click the result list pane item and selecting “Visualize tree”:

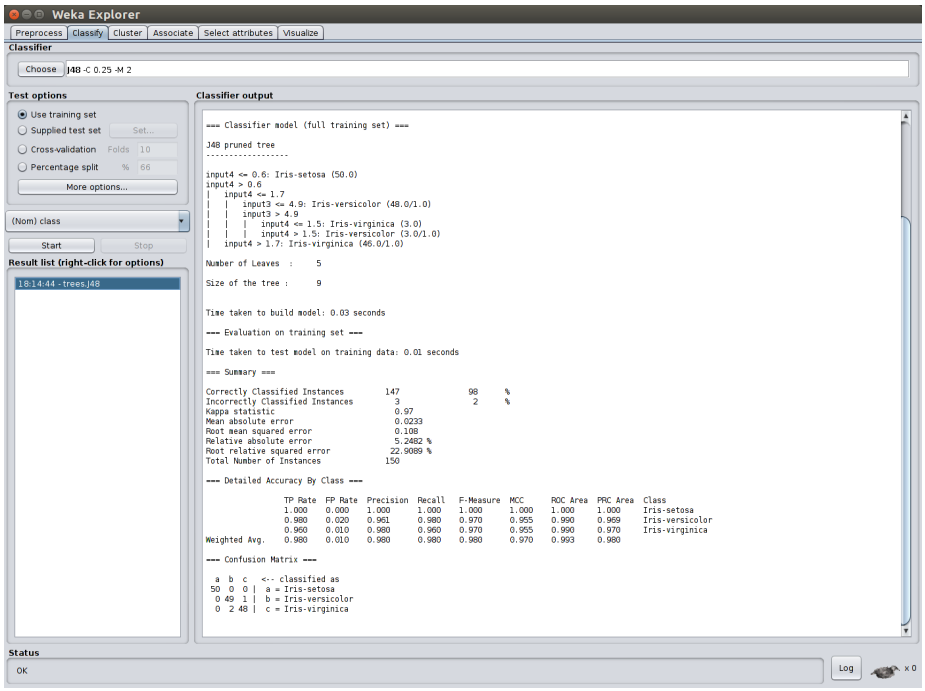


Figure 10: Output using J48 classifier

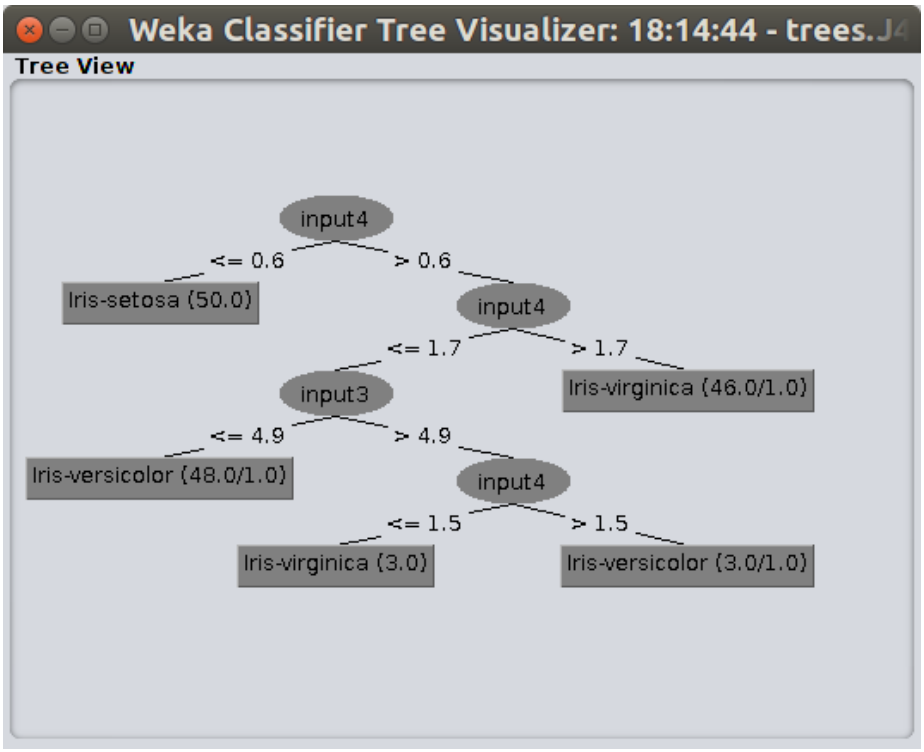


Figure 11: Decision tree for iris dataset according to J48

The decision tree shows step by step and with details the attributes that should be compared when trying to separate the items into the different classes.

Once we finished analyzing the input data we can now remove the class column from our input data and try to analyze it using only unsupervised tools.

2.1 Using Principal Component Analysis (PCA)

In order to use the principal component analysis, I first changed the dataset and remove the class column:

Figure 12: Remove class column from dataset

Then we go to Weka, import new dataset, go to tab “Select attributes”, then go to “Select Evaluator” section and choose “PrincipalComponents”, leave the “Search Method” as default and then we press start button.

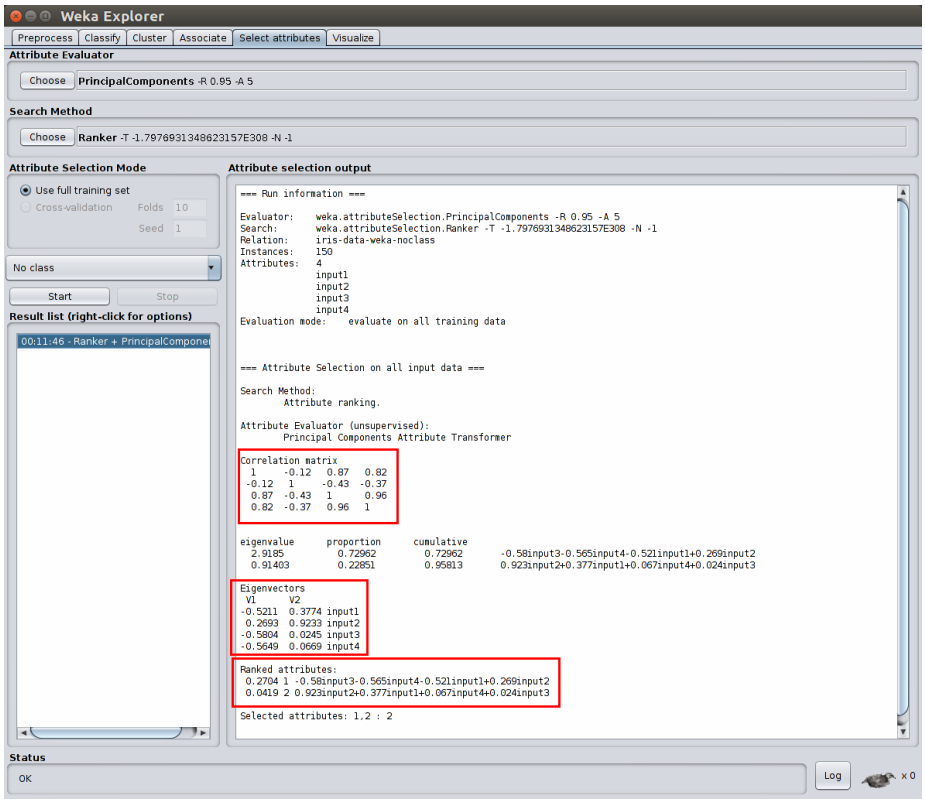


Figure 13: PCA results for no-class dataset

Maybe the must important and useful information from PCA is the ranked importance for the attributes, it means, how many separable or variance information is allocated in a certain attribute from the dataset. In this case, the combination of attributes showed in the first line after "Ranked attributes" has a significant importance of 27.04%.

Besides we can go back to "Preprocess" tab and there apply a PrincipalComponents filter to our data:

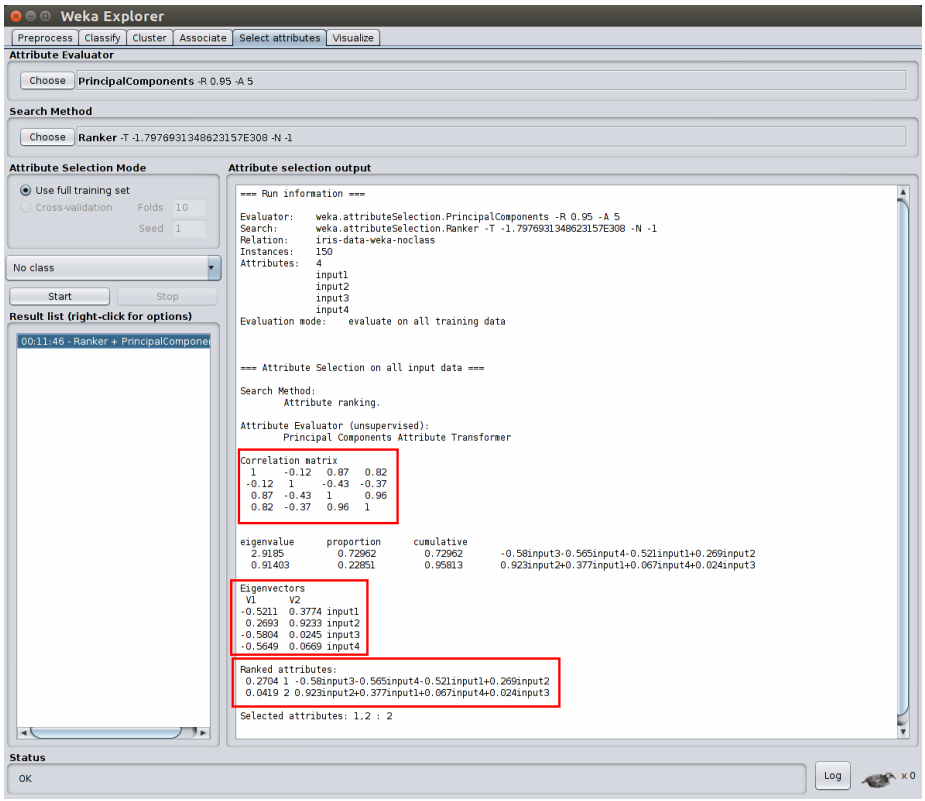


Figure 14: Preprocess before applying the filter

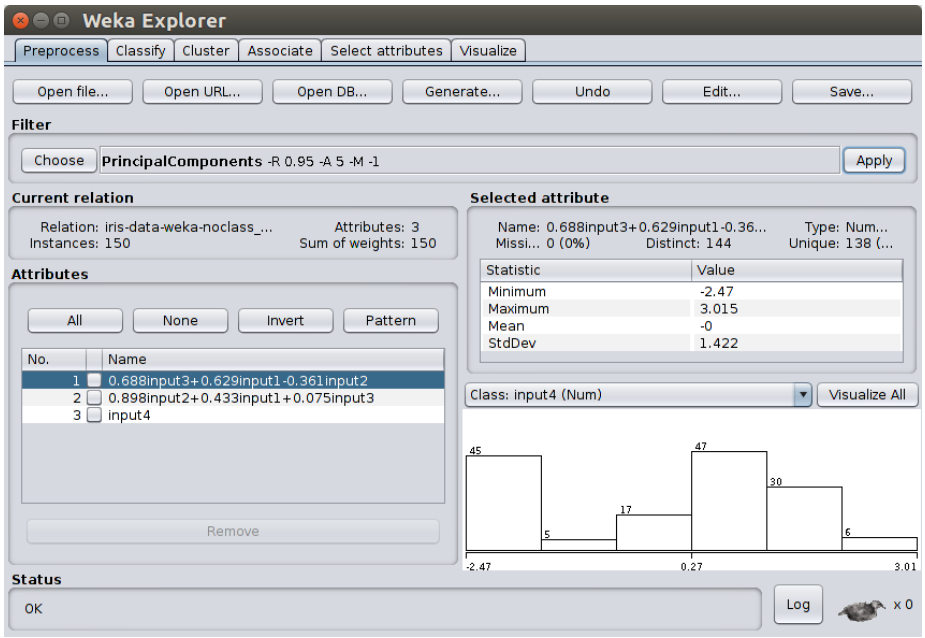


Figure 15: Preprocess after applying the filter

After we apply the filter, then we can go to "Visualize" tab and there watch a detail of the variance in a plot:

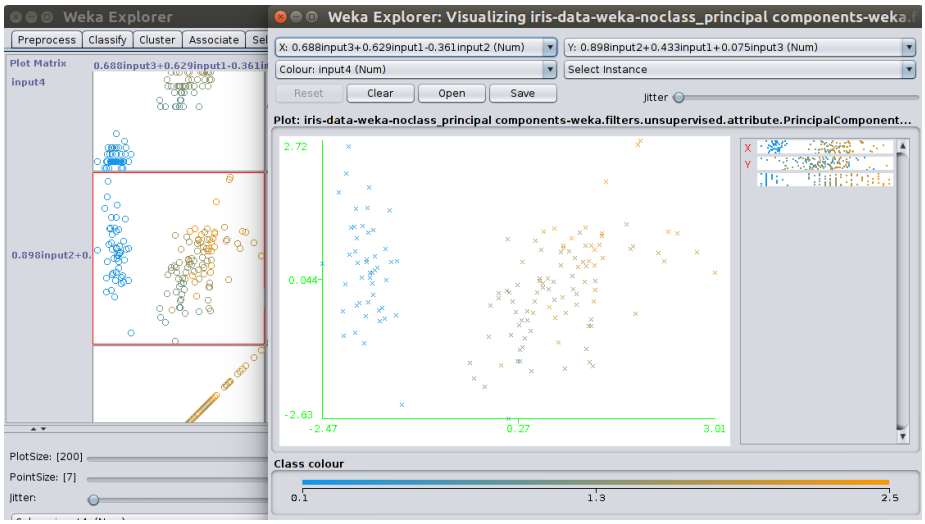


Figure 16: PCA plot

2.2 Using K-means

To use K-means in Weka we simply must go to "Cluster" tab, choose "SimpleK-Means" analyzer in the "Clusterer" section, and then just change the property numClusters in order to change the number of "k".

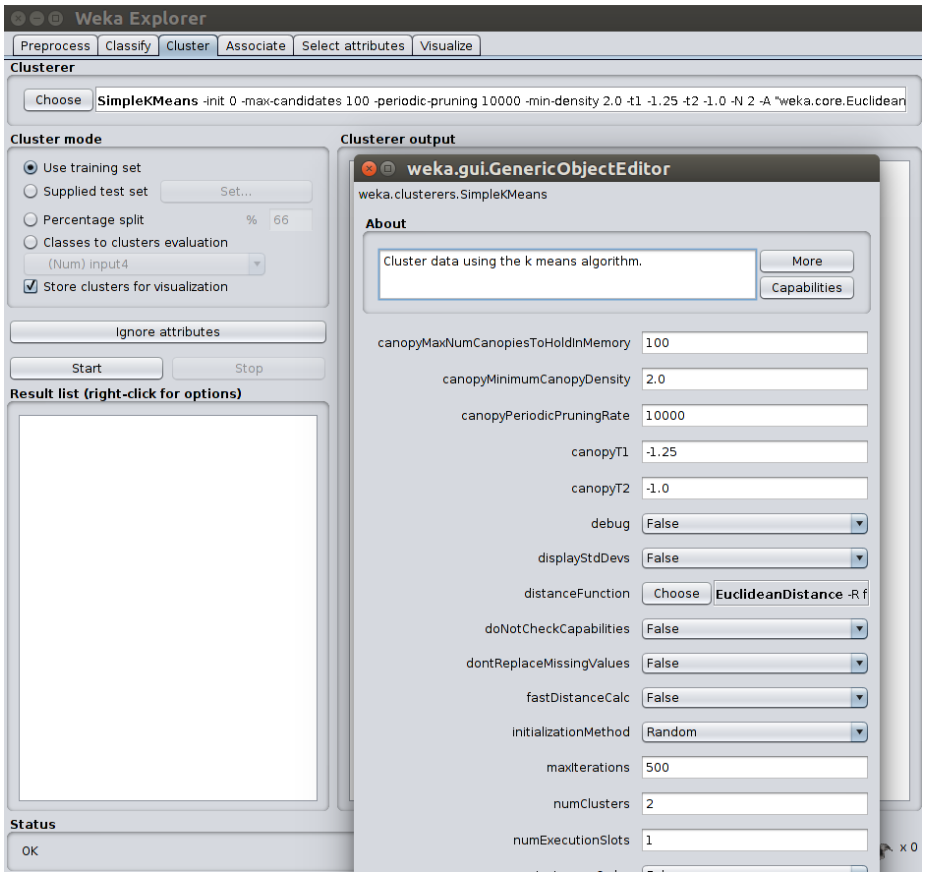


Figure 17: K-means in Weka

Now lets check the results for every number of means:

2.2.1 k=2

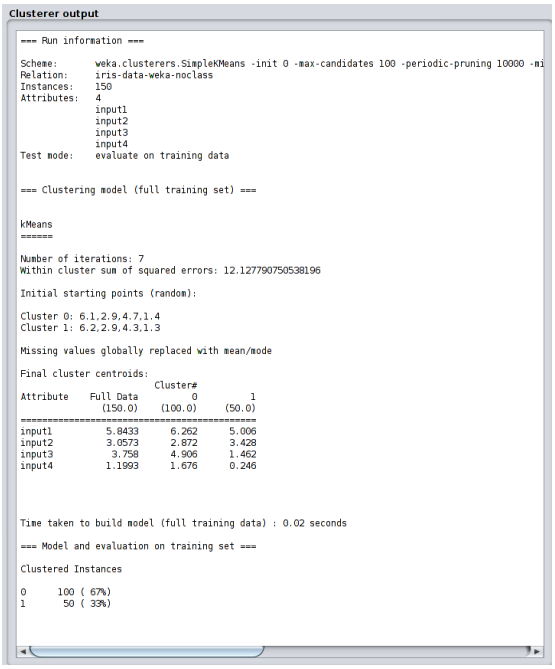
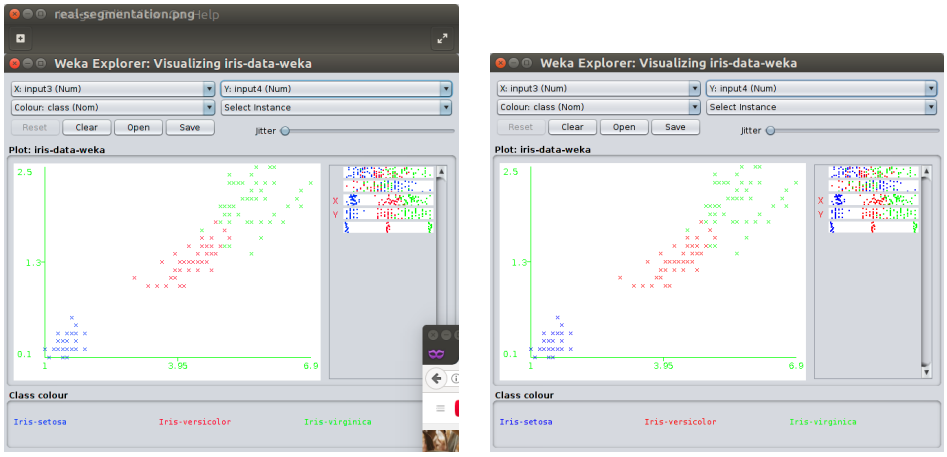


Figure 18: Logger result for K-means using k=2



(a) (b)

Figure 19: Comparison k=2 and original classification

2.2.2 k=3

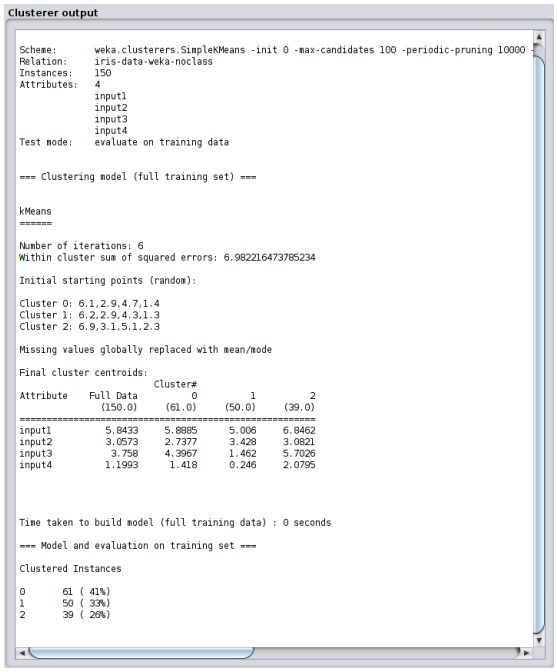


Figure 20: Logger result for K-means using k=3

(a) (b)

Figure 21: Comparison k=3 and original classification

2.2.3 k=4

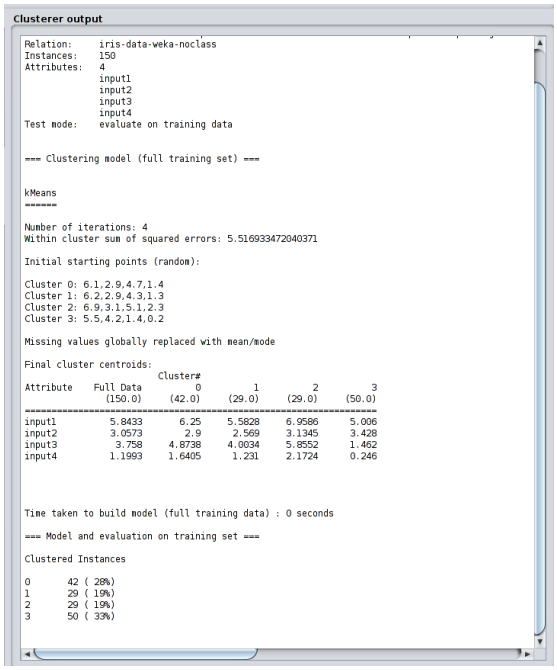
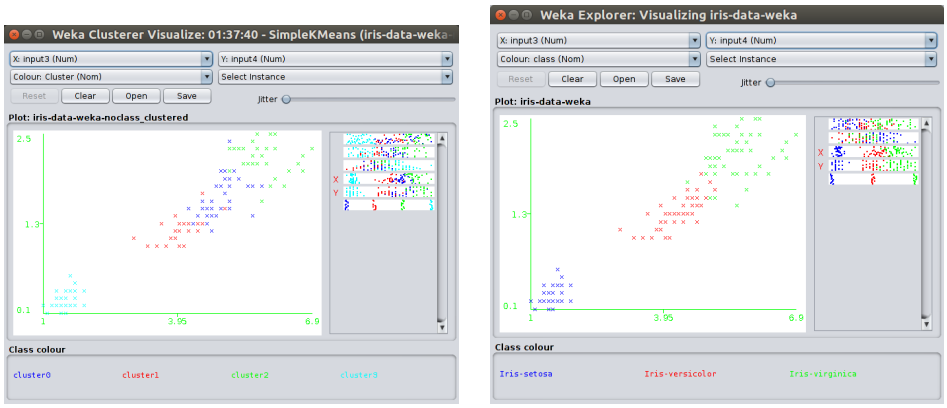


Figure 22: Logger result for K-means using k=4



(a) (b)

Figure 23: Comparison k=4 and original classification

2.2.4 k=5

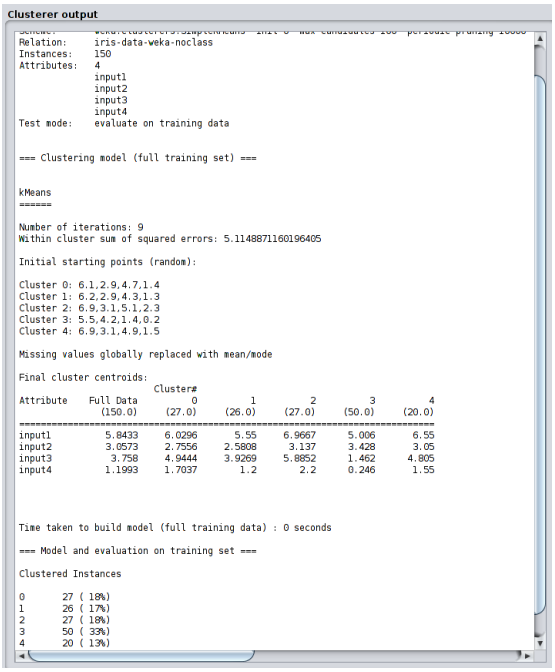
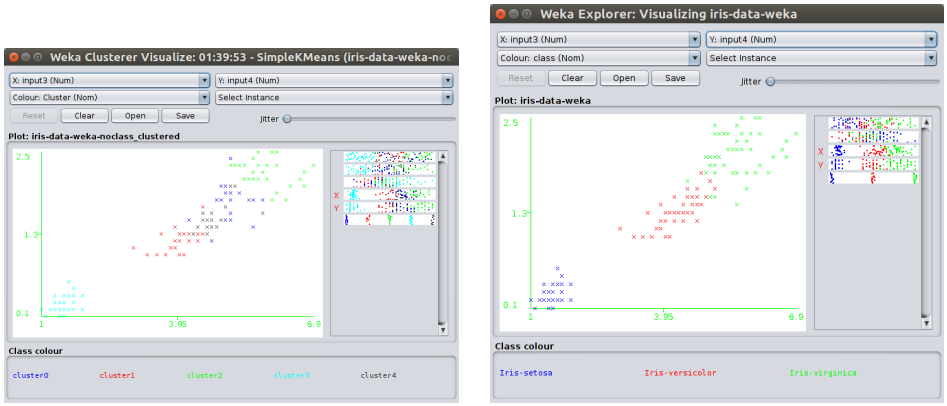


Figure 24: Logger result for K-means using k=5



(a) (b)

Figure 25: Comparison k=5 and original classification

2.3 Using Self-Organizing Maps (SOM)

Before start working with self-organizing maps analysis in Weka, we first must install the package using the package manager:

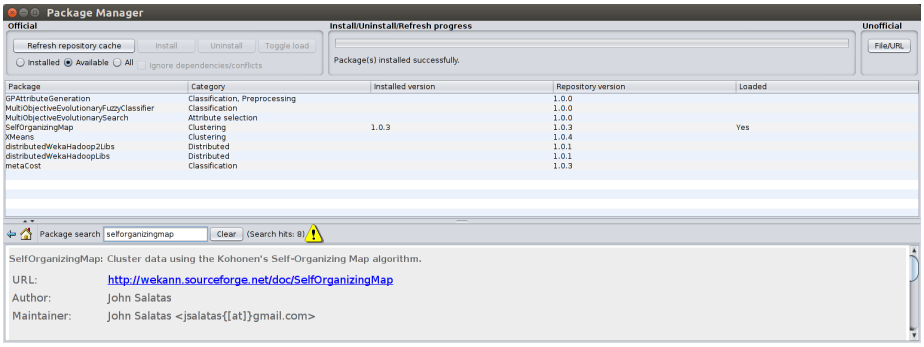


Figure 26: Installing self-organizing maps package in Weka [**?**]

When working with self-organizing maps, the main idea is trying is trying with different settings and check the results to try finding the one which better fit our real data clusters.

Then we first try using convergenceEpochs=1000, height=2, leaningRate=1.0, orderingEpochs=2000, width=2.

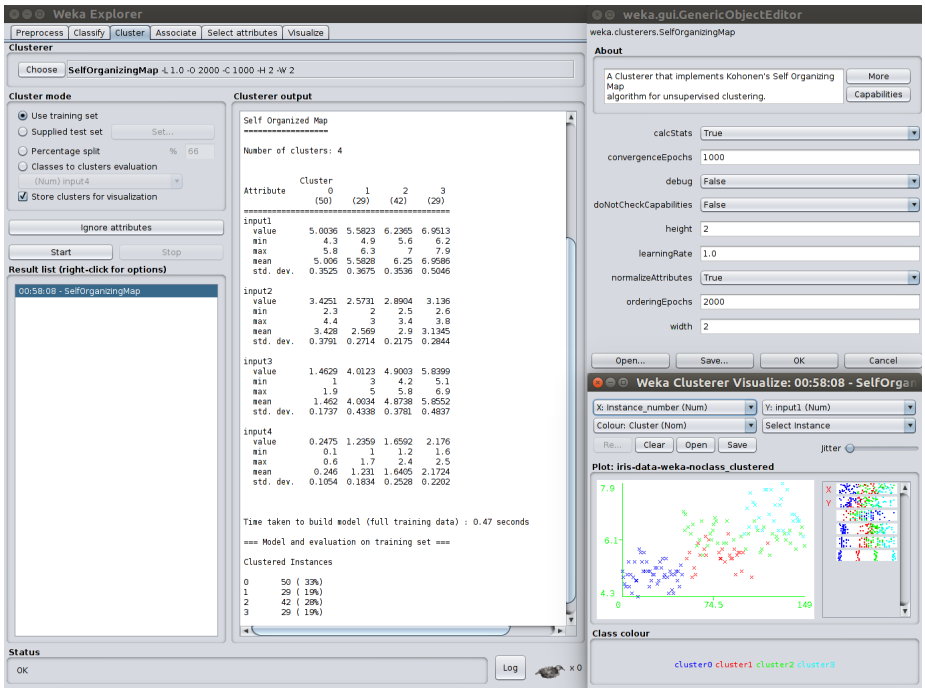


Figure 27: SOM, attempt 1

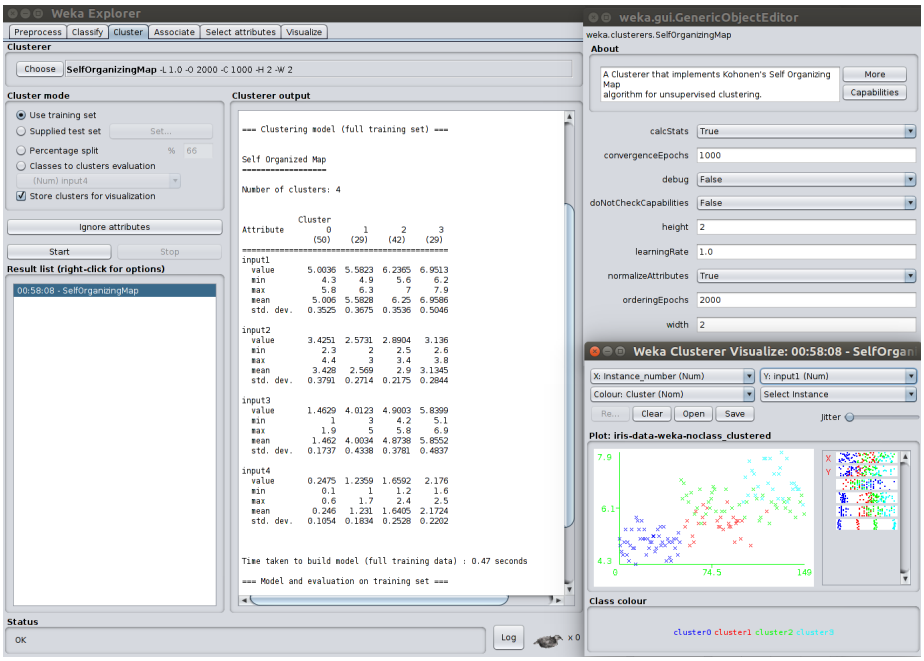


Figure 28: SOM, attempt 2

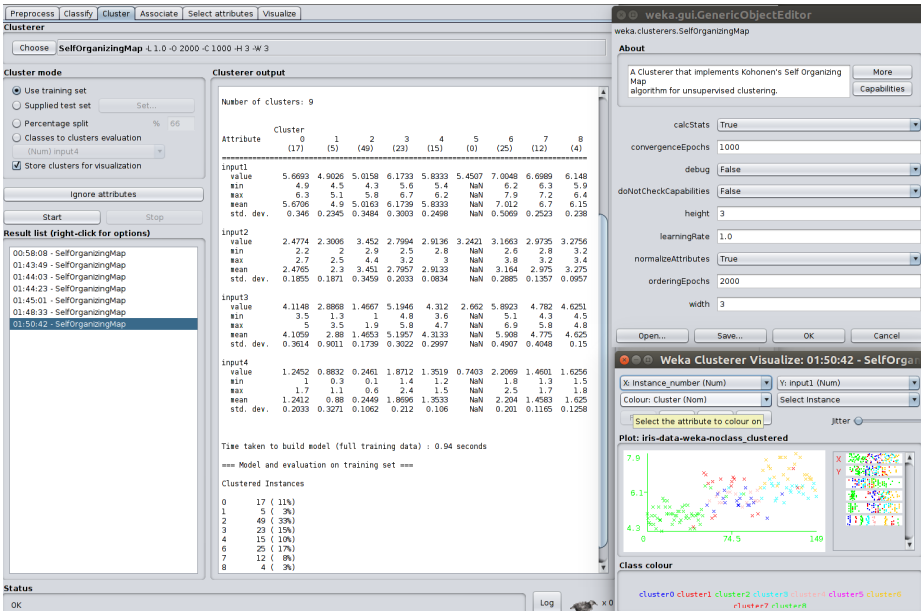


Figure 29: SOM, attempt 3

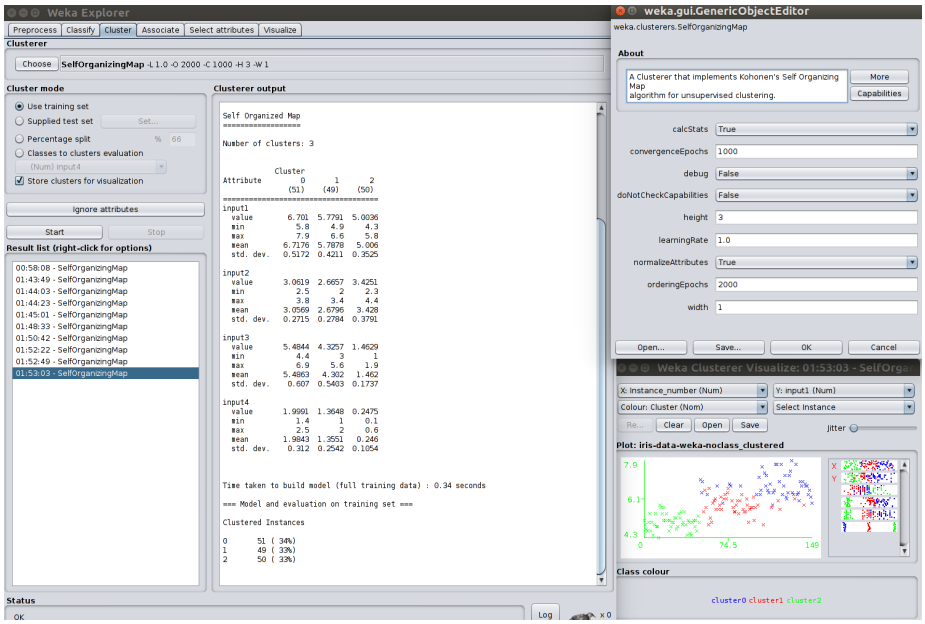


Figure 30: SOM, attempt 4

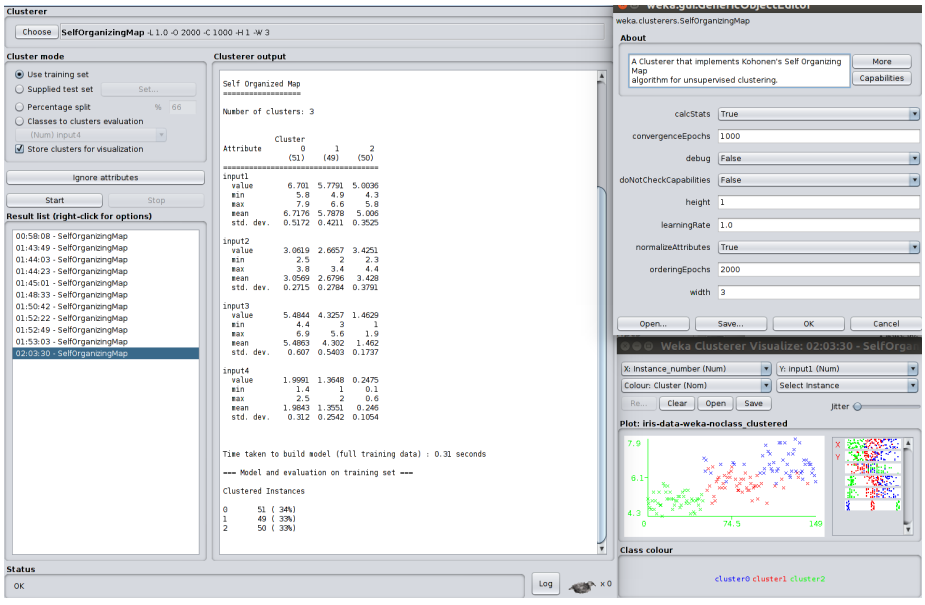


Figure 31: SOM, attempt 5

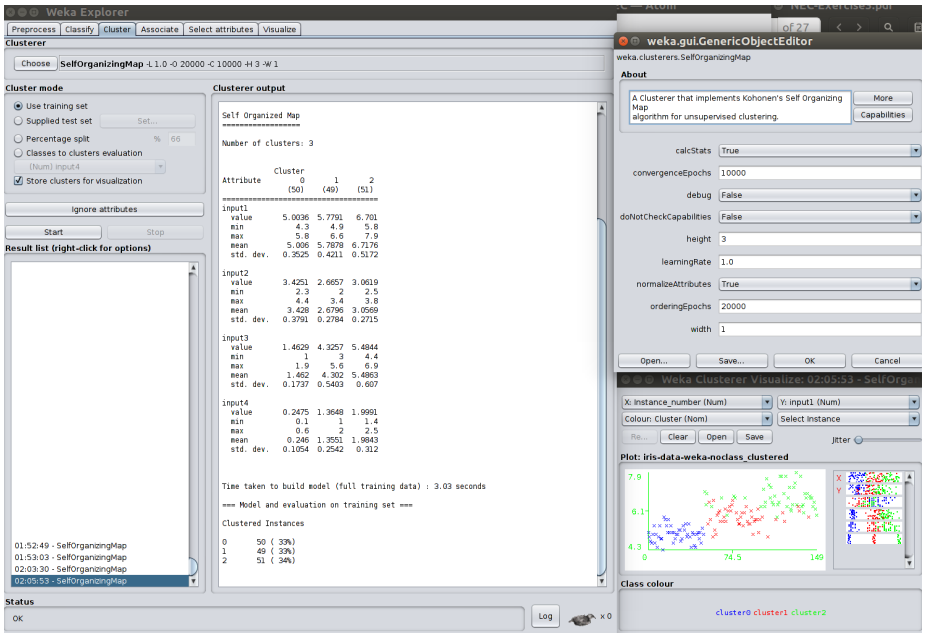


Figure 32: SOM, attempt 6

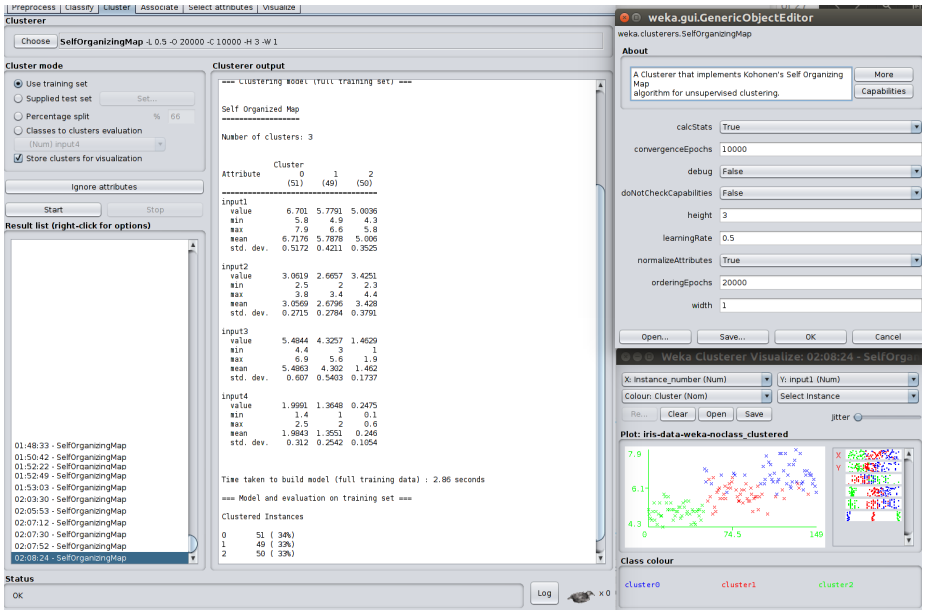


Figure 33: SOM, attempt 7

Just checking the result when using big numbers for convergence and order-

ing epochs.

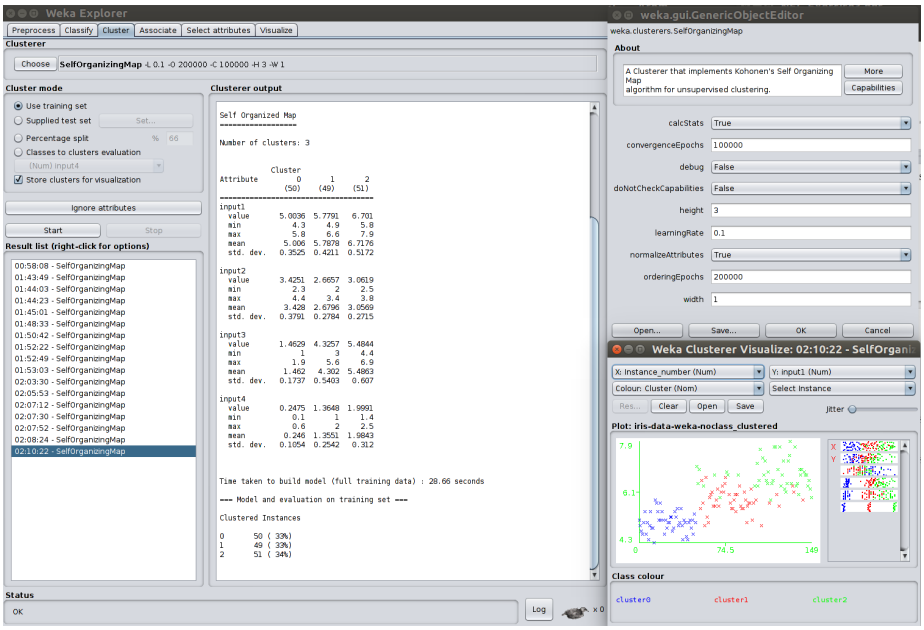


Figure 34: SOM, attempt 8

I think the best results where always using height=3, width=1 and learningRate=0.1. Changing the other parameters didn't affect the results very much.

Lets check the output plot for this "best" in more detail:

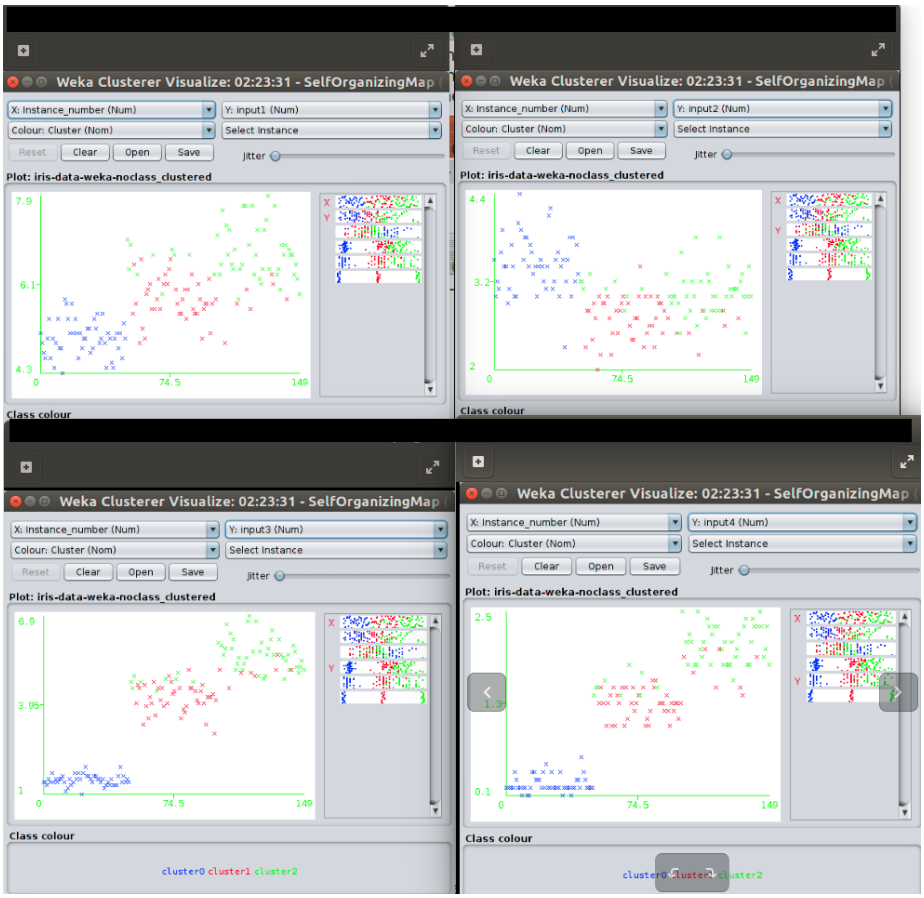


Figure 35: SOM, Detail

3 Comparison of results

The PCA projection are a very useful tool to have a good idea about which attributes have the most relevance, once we know that is much easier to analyze our data and understand the results.

Our results when using K-means were useful and very accurate, but is always very dependent in the number of classes (means) we define, otherwise the results are away from those reals. Maybe it would be better having more parameters and this way we could use them to make this means-dependence a little more adjustable.

SOM algorithm return much more useful information than just the clusters and the final classification, with all the information about the attributes we also can have a description of the distribution of the samples, including minimums, maximums, means and standard deviations.

References

[1] Wikipedia. Iris flower data set — wikipedia, the free encyclopedia, 2017. [Online; accessed 3-February-2017].

[2] Wikipedia. Principal component analysis — wikipedia, the free encyclopedia, 2017. [Online; accessed 3-February-2017].

[3] Wikipedia. K-means clustering — wikipedia, the free encyclopedia, 2017. [Online; accessed 3-February-2017].

[4] Wikipedia. Self-organizing map — wikipedia, the free encyclopedia, 2017. [Online; accessed 3-February-2017].