Orysya Stus

Data Mining for Scientific Applications

Course No. CSE-40770

Laboratory Assignment II:

To download additional .arff data sets go to:

http://repository.seasr.org/Datasets/UCI/arff/
or alternatively
http://www.hakank.org/weka/

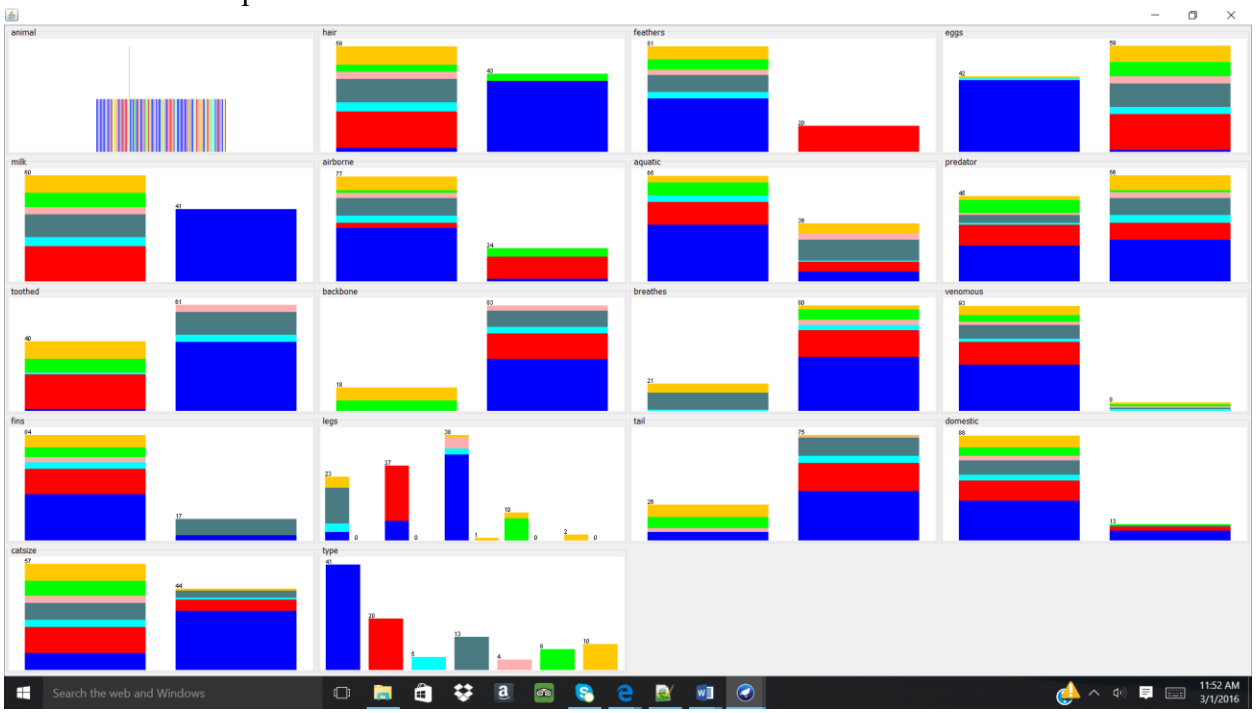
1. Use the following learning schemes to analyze the zoo data (in zoo.arff):

   OneR - weka.classifiers.OneR
   Decision table - weka.classifiers.DecisionTable -R
   C4.5 - weka.classifiers.j48.J48
   K-means - weka.clusterers.SimpleKMeans

   Try using reduced error pruning for the C4.5. Did it change the produced model? Why?

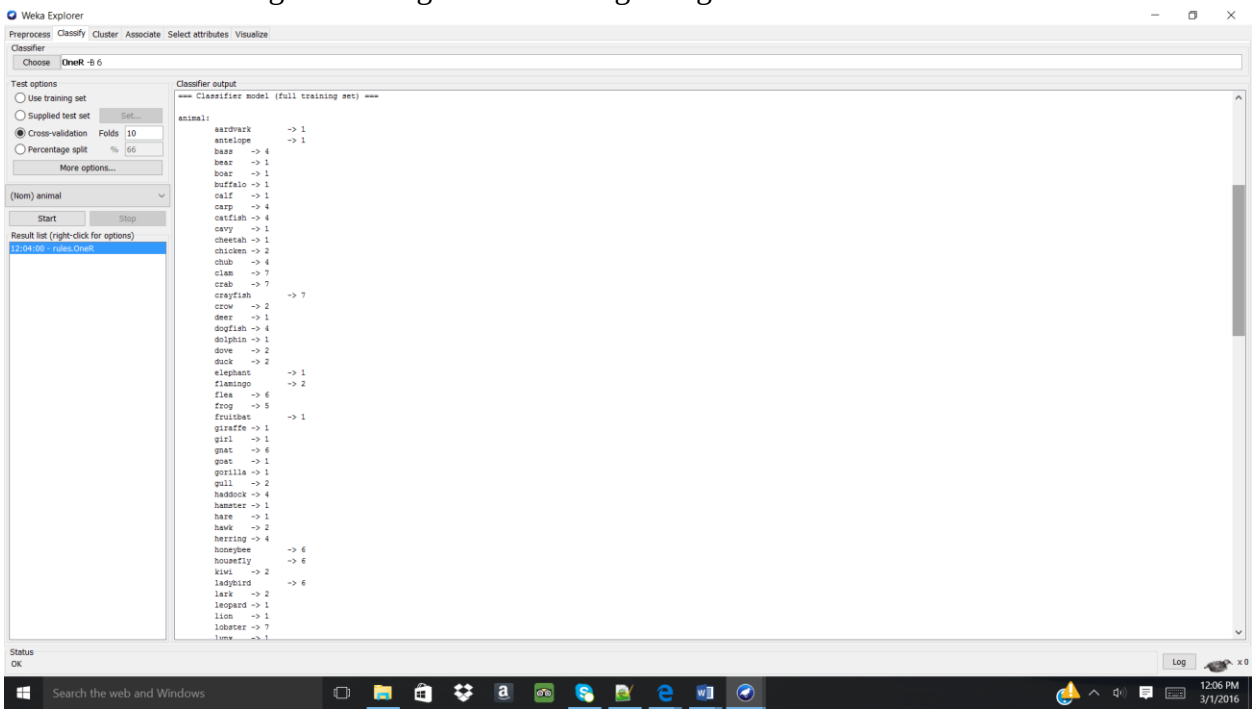
   For K-means, for the first run, set ***k***=10. Adjust as needed. What was the final number of ***k***? Why?

A. The zoo.arff dataset was opened in Weka and "Visualize All" was used to visualize the instance distributions for each of the 18 attribute values. Note: all attribute values are nominal values, and attribute "animal" provides a name to each individual instance.



Orysya Stus

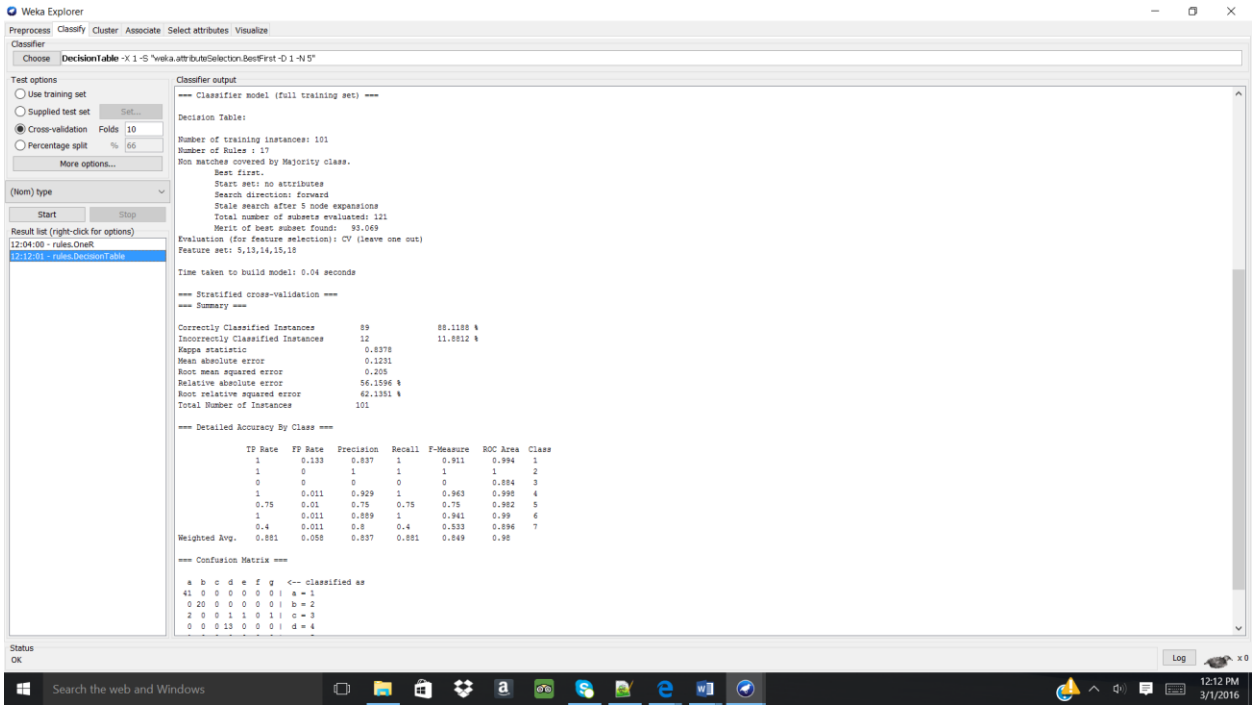
B. Classify/Classifier/rules/OneR was selected. Under “Test options”, “Cross-validation” at “Fold 10” was selected in order to accomplish the most believable evaluations where the data is split into training and test data sets 10 times with models being created, iterated, and polished between each training and testing. The following was generated:



Note: the class attribute is “type”. The algorithm creates a model based on a single attribute, to determine the classes for the datasets. The attribute “animal” is selected as providing the most information gain for the data and the OneR algorithm uses numeric restrictions in determining which instances fall under the class “type”. This rule maps 43 instances correctly, but 58 incorrectly with a mean absolute error of 0.1641. Although more information is learned from the OneR rule in comparison to the ZeroR rule, error still exists in the system showing that a single rule based on one attribute cannot map out all of the instances properly, therefore a more complex rule/model needs to be used.

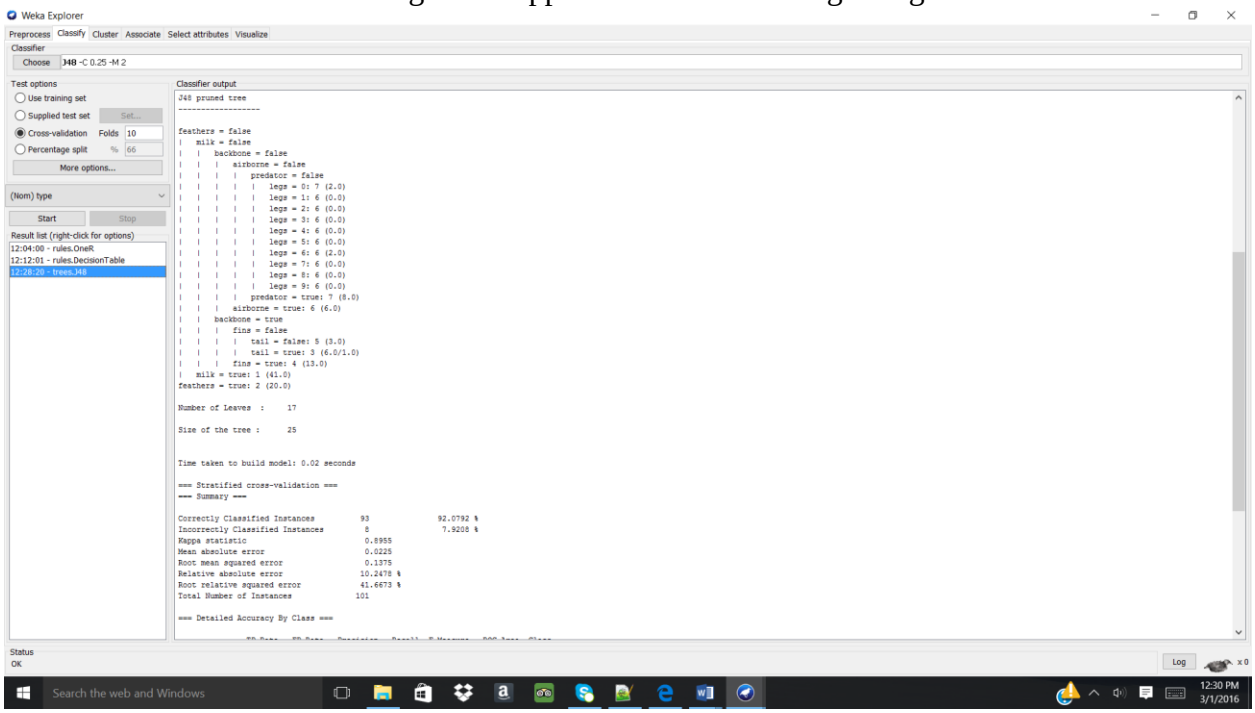
C. Classify/Classifier/rules/DecisionTable was selected. Under “Test options”, “Cross-validation” at “Fold 10” was selected in order to accomplish the most believable evaluations where the data is split into training and test data sets 10 times with models being created, iterated, and polished between each training and testing. The following was generated:

Orysya Stus

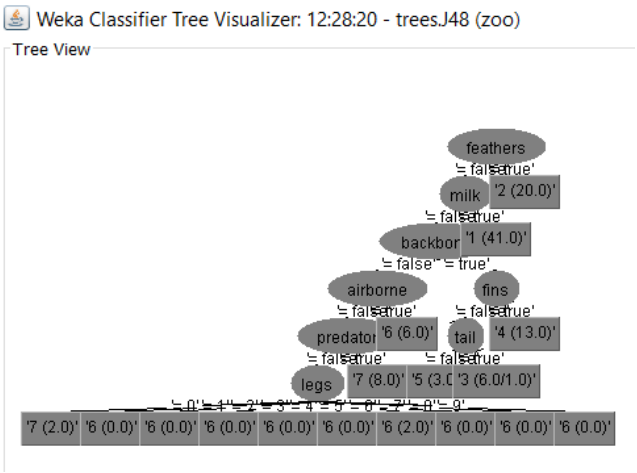


17 rules were generated with 89/101 instances classified correctly. The mean absolute error was lower compared to the OneR, at 0.1231.

D. Classify/classifiers/trees/J48 was selected. Under “Test options”, “Cross-validation” at “Fold 10” was selected. Default settings were applied and the following was generated:

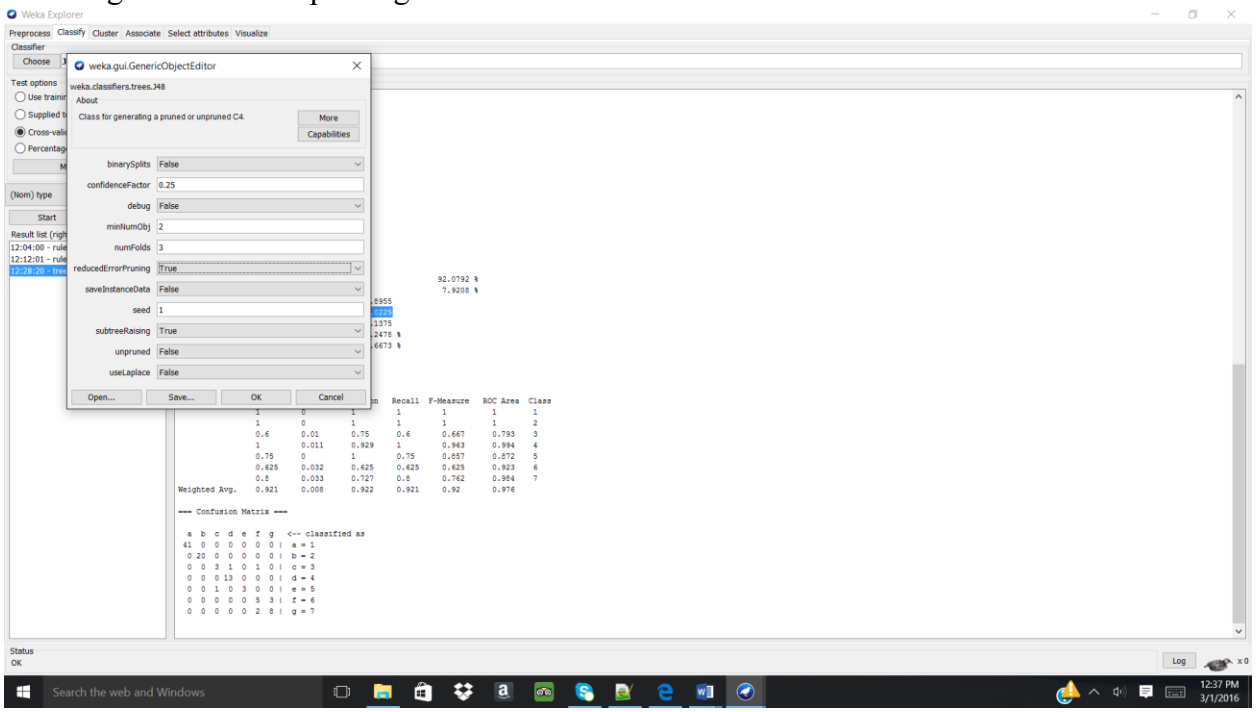


Orysya Stus



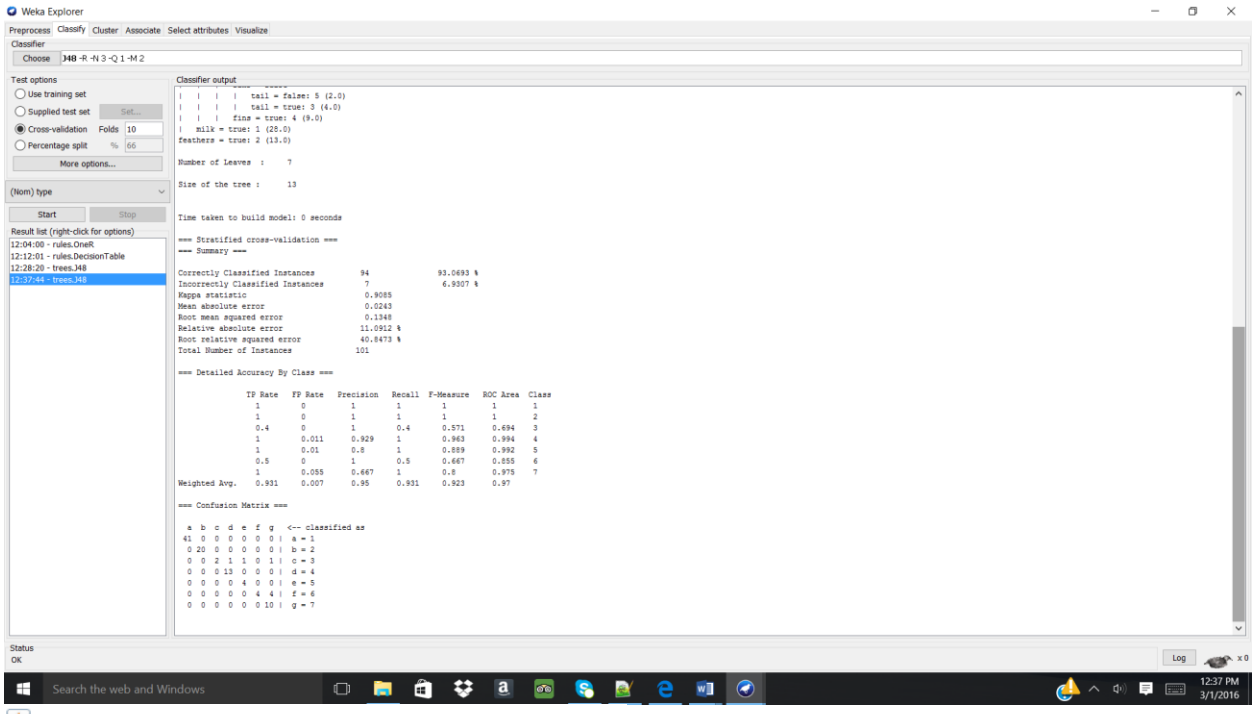
Here, the attribute “feathers” provides the most information gain in determining the class attribute “type”. The tree generated was able to classify 93/101 instances correctly, with a mean absolute error of 0.0225. Thus, the generated decision tree provides the most accurate information gain in classifying the data set.

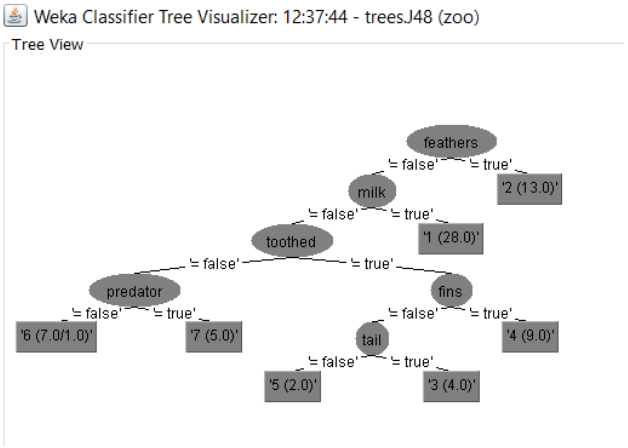
Selecting reduced error pruning to “True”:



Generated the following decision tree:

Orysya Stus



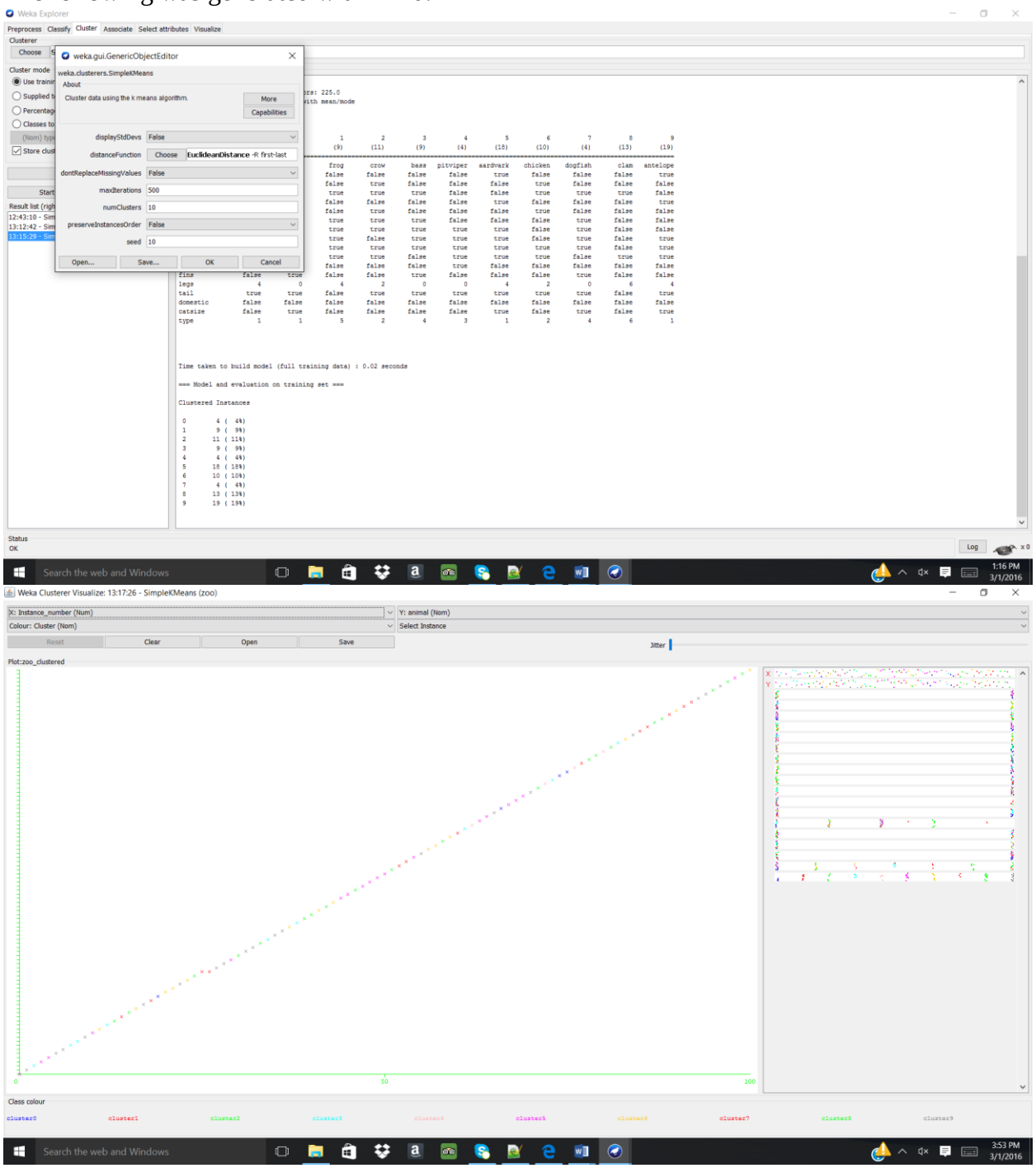


Compared to the tree generated above, selecting reduced error pruning creates a cleaner model which actually correlates to 94/101 classified correctly with a mean absolute error of 0.0243. Although the error is slightly larger, the model works well.

E. Cluster/SimpleKMeans was selected. Under test options, “Full training data” was selected in order to use a substantial amount of training data to generate a model which can be tested upon.

Orysya Stus

The following was generated with k=10:



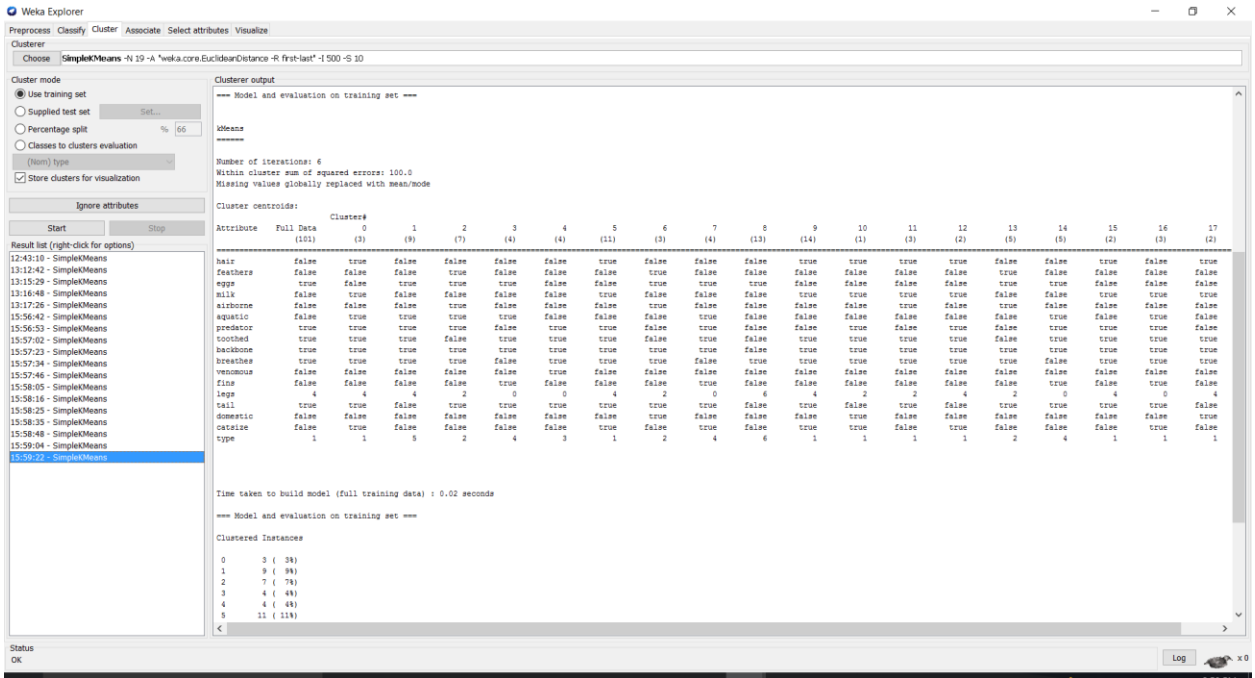
When working with k-means, choosing the best number of clusters, k, can be completed by varying k through multiple iterations and determining which value of k results in the lowest error.

When:

k = 1 sum of squared errors = 134.0

k = 19 sum of squared errors = 100.0

Orysya Stus



K= 19 was choosen for the k-clusters since in the data set there are 19 divisions of the attribute “animal”.

2. Use the following learning schemes to analyze the Automobile (http://archive.ics.uci.edu/ml/datasets/Automobile) data set.

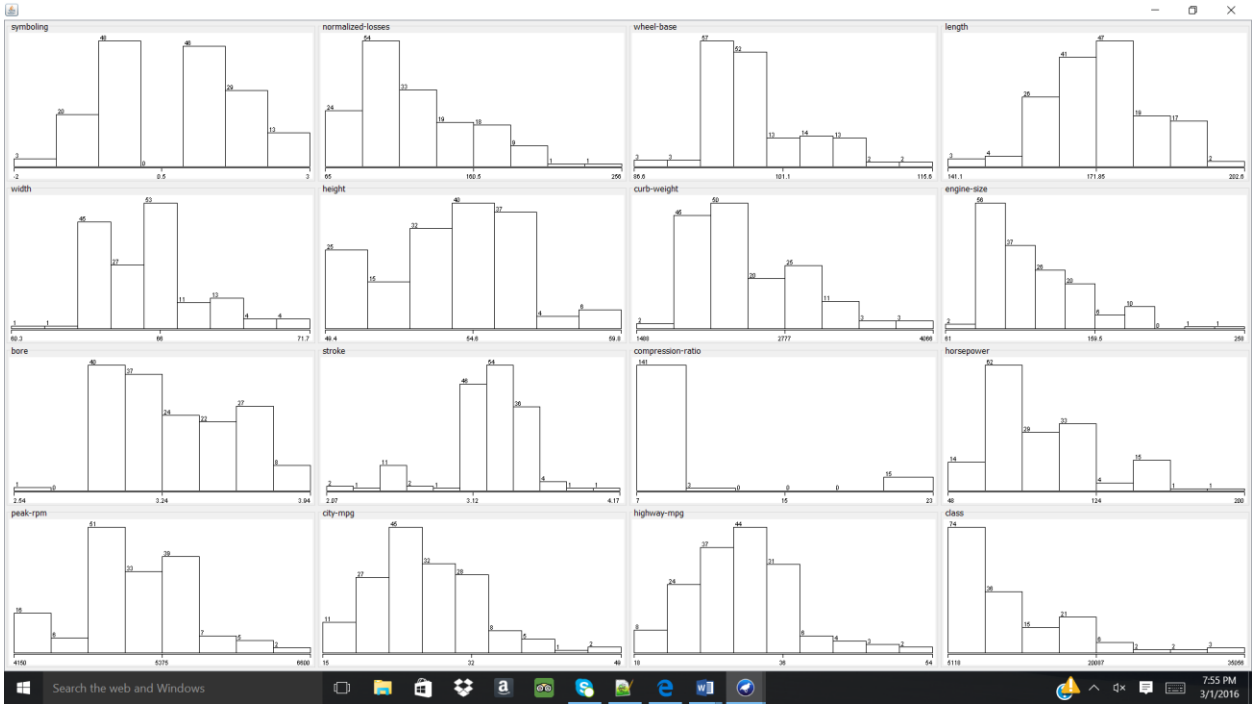
Linear regression	- weka.classifiers.LinearRegression
M5'	- weka.classifiers.M5'
Regression Tree	- weka.classifiers.M5'

Orysya Stus

K-means clustering - weka.clusterers.SimpleKMeans

A) How many leaves did the Model tree produce? Regression Tree? What happens if you change the pruning factor?
How many clusters did you choose for the K-means method? Was that a good choice? Did you try a different value for ***k***?

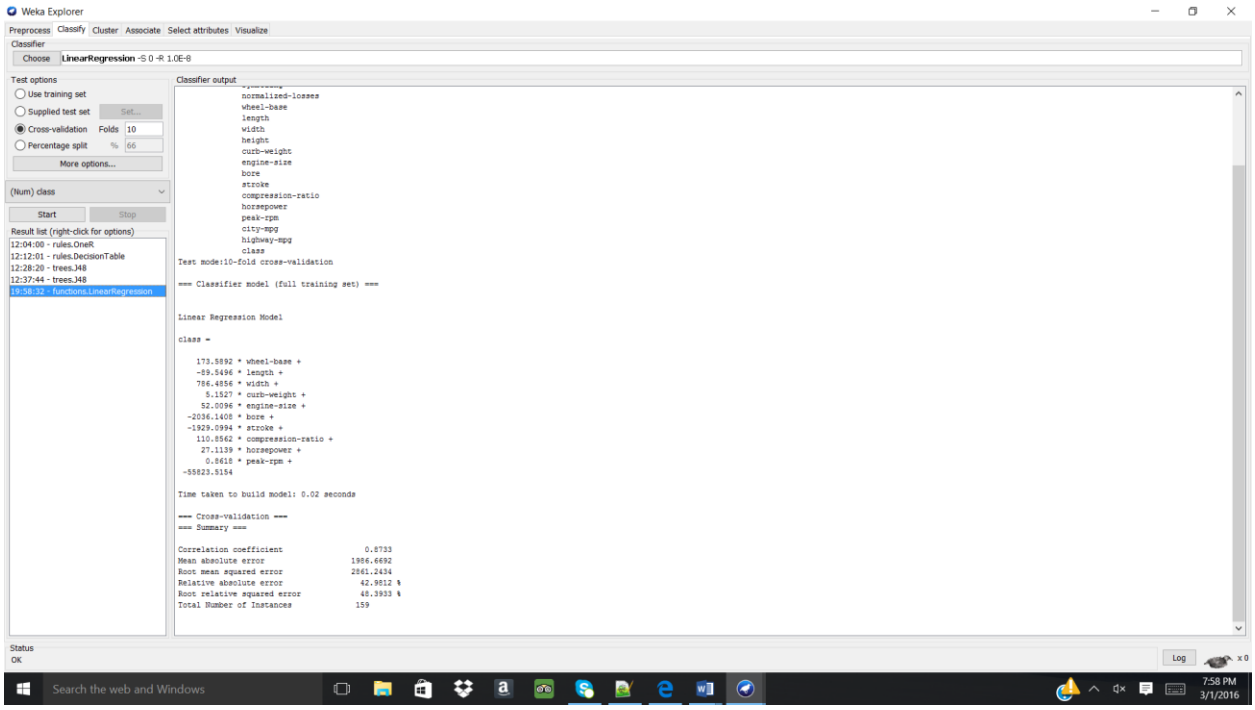
A. The file autoPrice.arff was opened in Weka and "Visualize All" produced the following:



Note: all attribute values are numeric and the class attribute "class" correlates to the price of the vehicle.

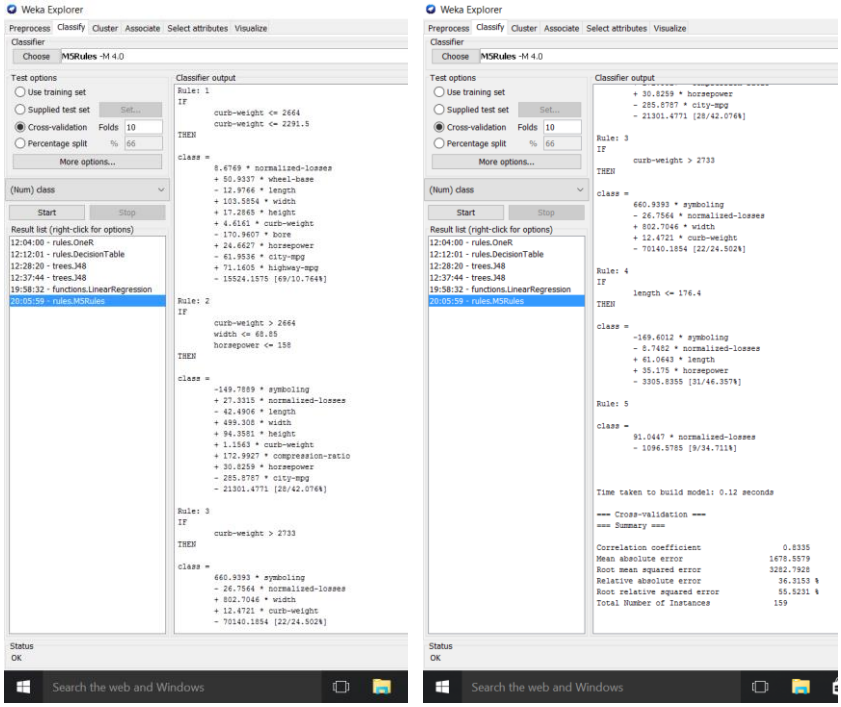
B. Classify/functions/LinearRegression was selected. The "test option", "Cross-validation Fold 10" was selected. Further default values were maintained, and the following was generated:

Orysya Stus



The generated equation assigns differnet weights to the attribute values in combination in order to determine the value of the class, this equation has a correlation coefficient of 0.8733.

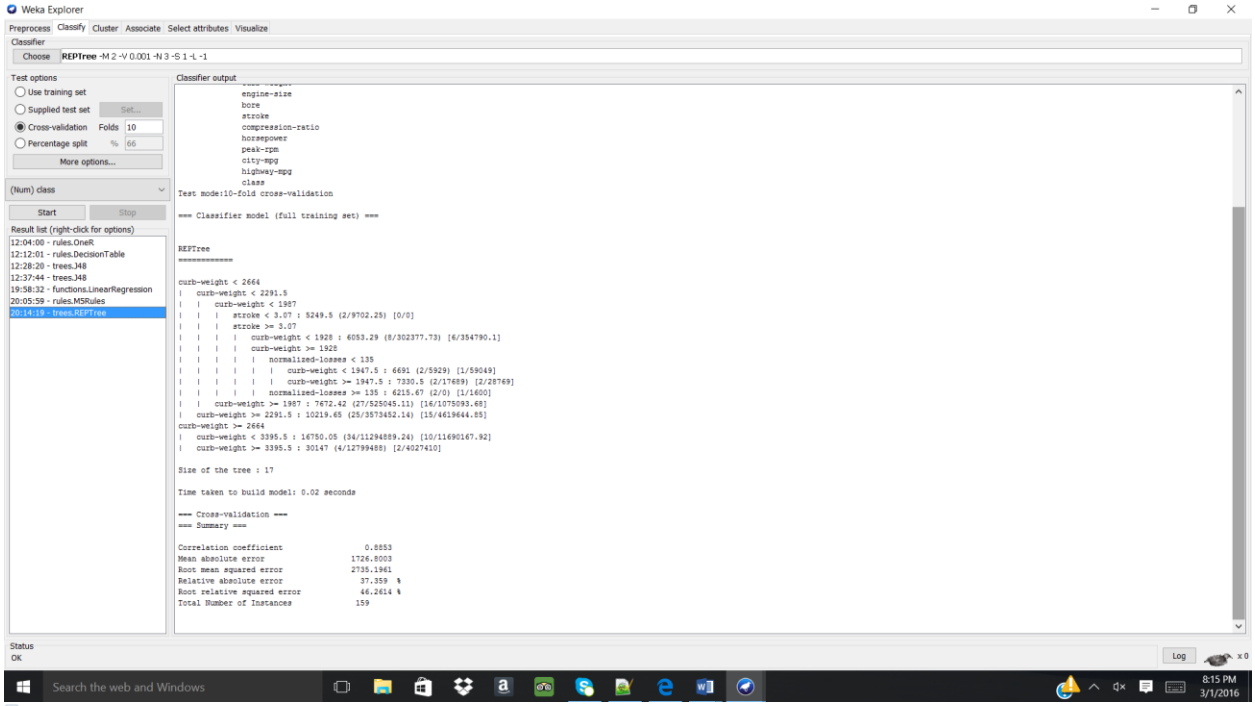
C. Classify/rules/M5Rules was selected with “Cross-validation Fold 10” as the test option. The following 5 rules were generated:

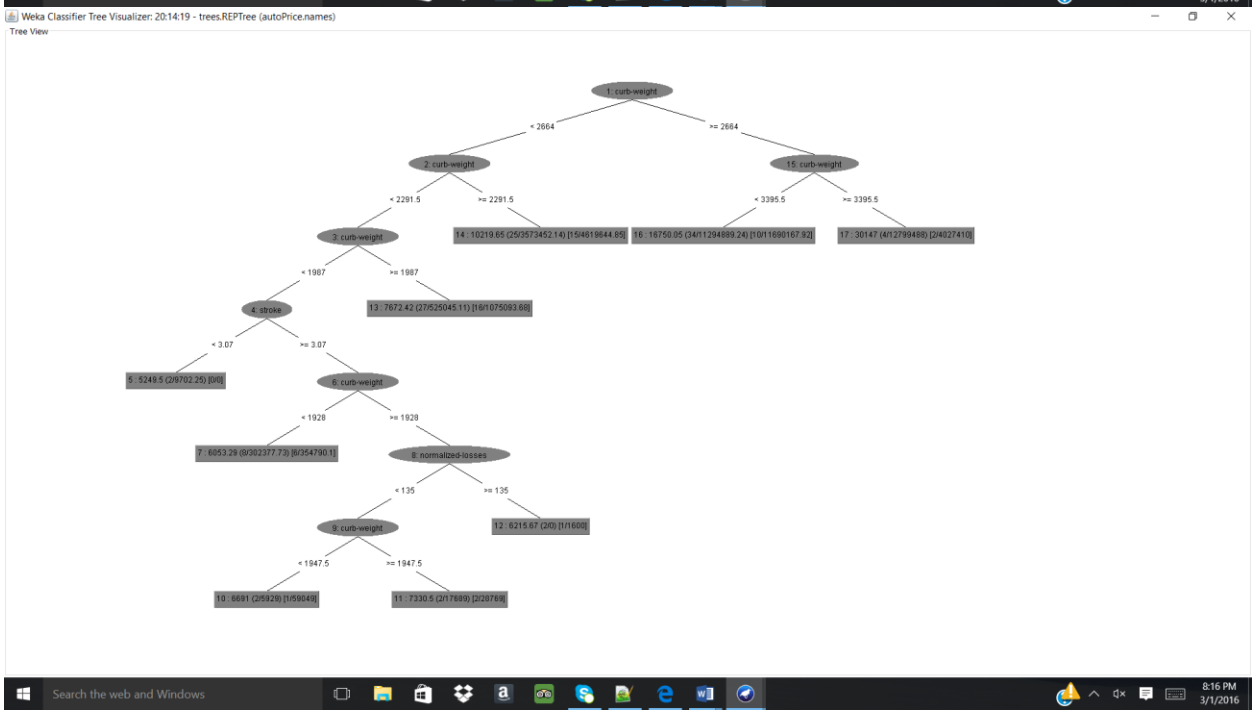


These rules are generated using “perfect” rules that use combinations of attribute values in order to find the value of “class”. The correlation coefficient is slightly lower compared to that of the linear regression equation, at 0.8335.

Orysya Stus

D. Classify/trees/REPTree was selected with “Cross-validation Fold 10” as the test option.

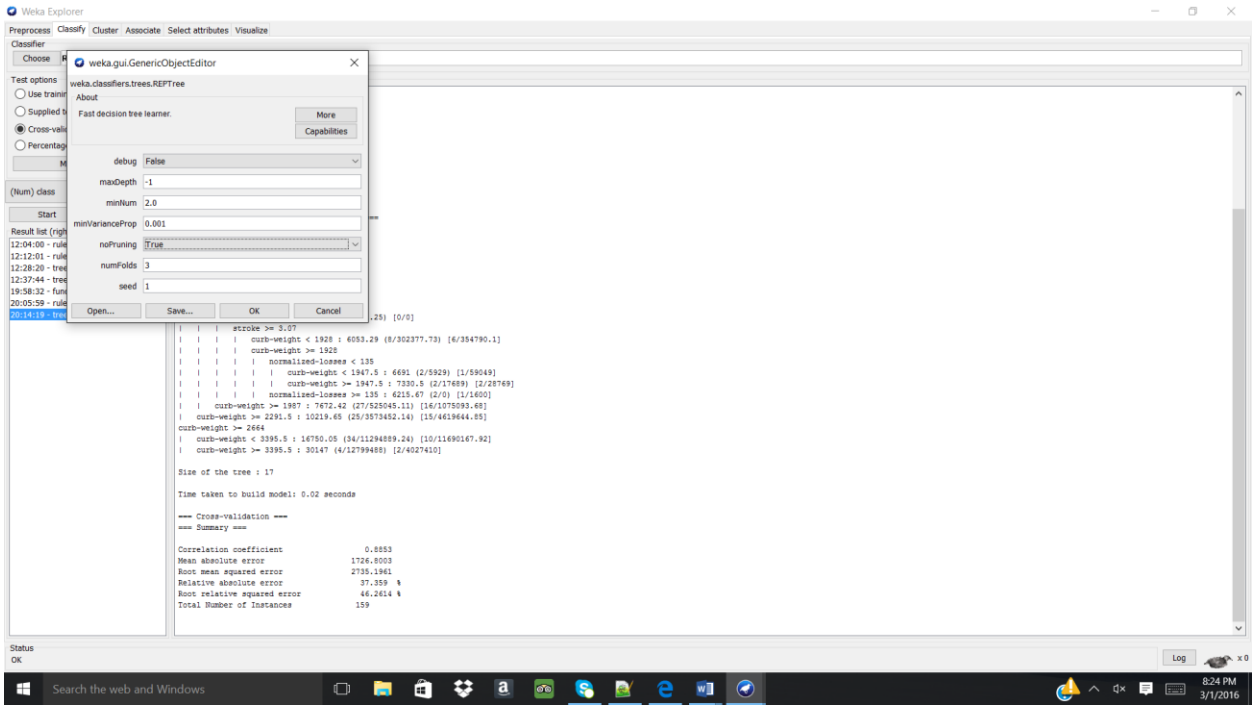




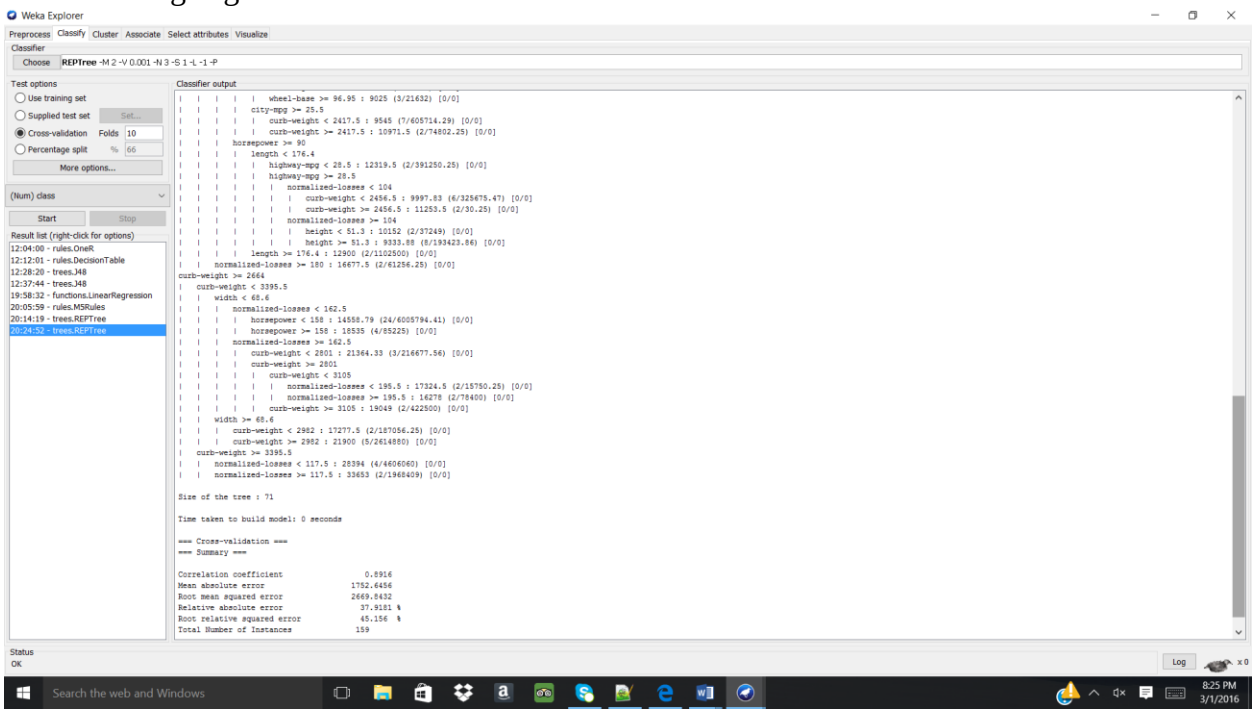
The tree generated has a higher correlation coefficient compared to the above learning schemes, at 0.8853. The model tree has 9 nodes.

When the pruning factor is changed where “nopruning” is changed from false to true:

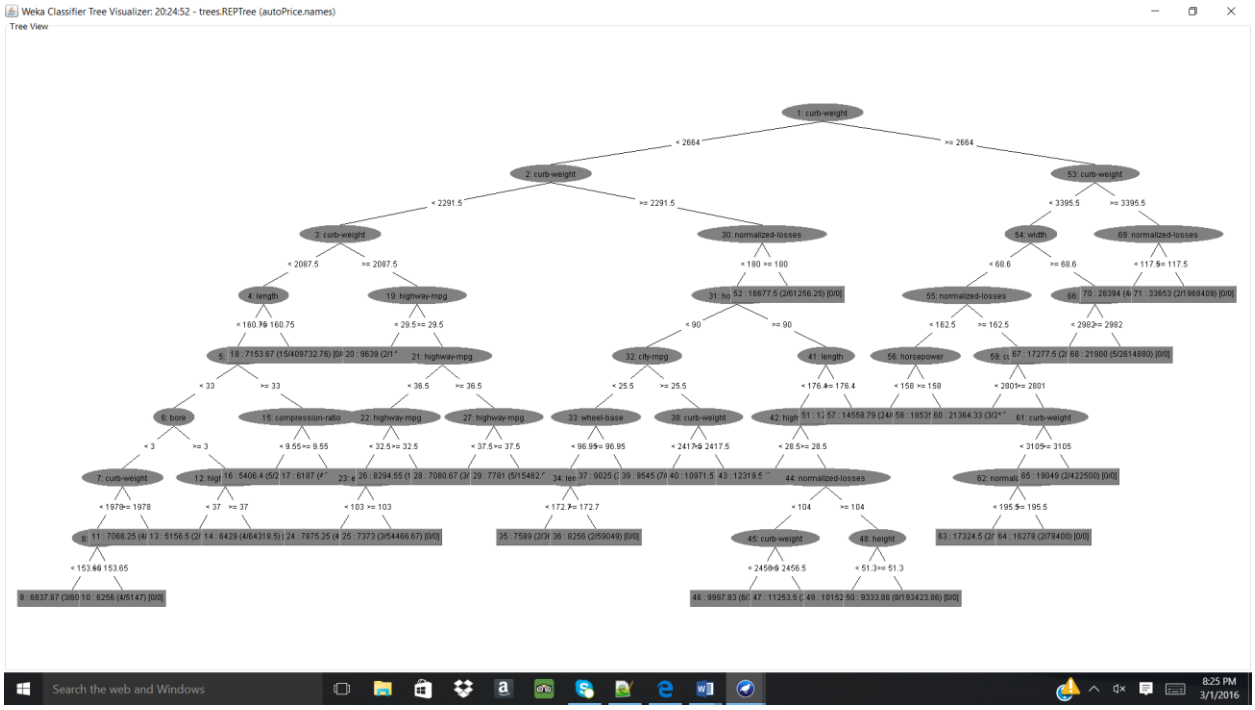
Orysya Stus



The following is generated:

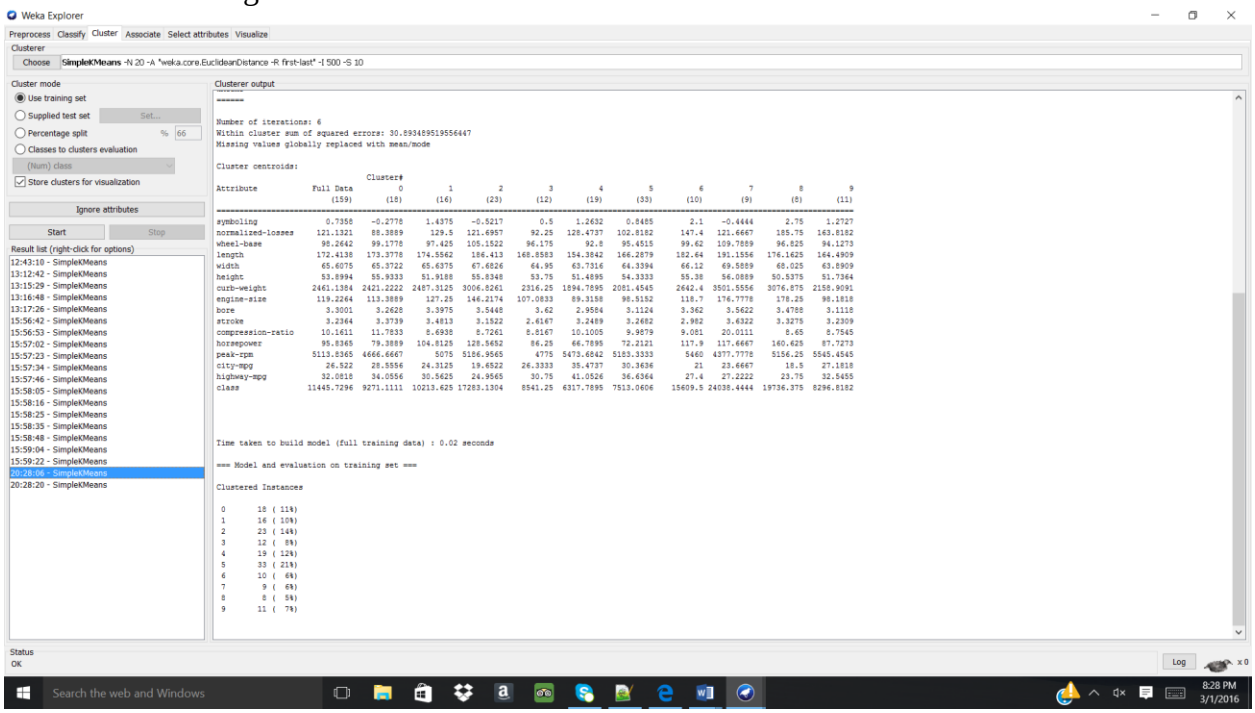


Orysya Stus



The correlation coefficient falls down to 0.8916 and the number of leaves greatly increases.

E. Cluster/clusterers/SimplekMeans was selected with the entire training set being used to generate the model. K = 10 was used for the number of clusters since this provided the lowest error in the learning scheme.

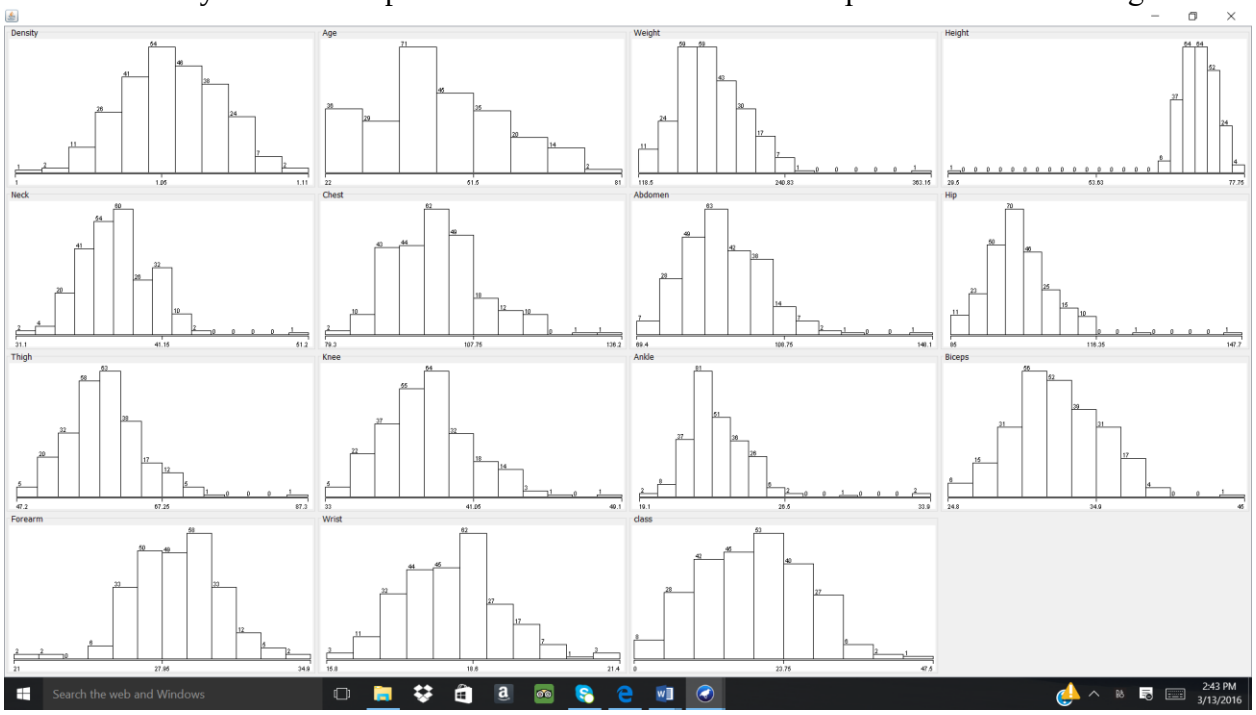


Taken together, the highest correlation coefficient belonged to the pruned model tree and allowed for prediction of the dataset the best. The attribute “curb_weight” was used provided the msot information gain to this dataset.

Orysya Stus

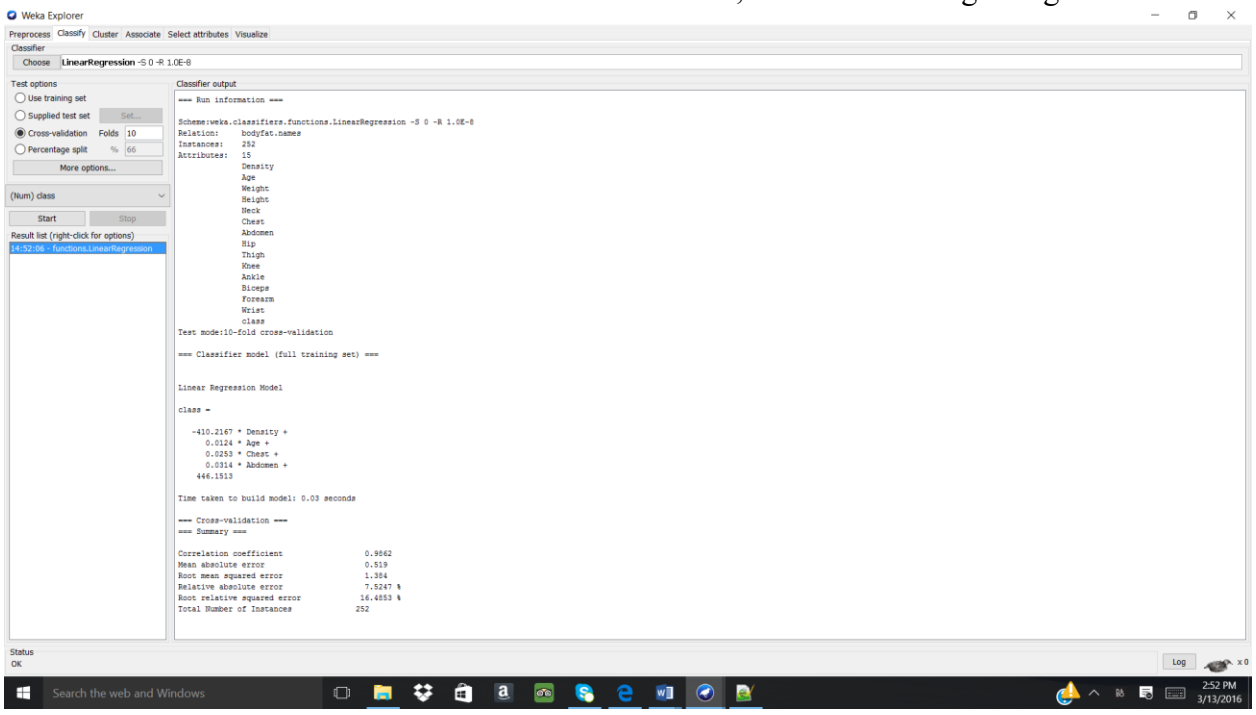
B) Now perform the same analysis on the bodyfat.arff data set.

A. The file bodyfat.arff was opened in Weka and “Visualize All” produced the following:



Note: all attribute values are numeric and the class attribute “class” correlates to the numeric value of the individual’s BMI.

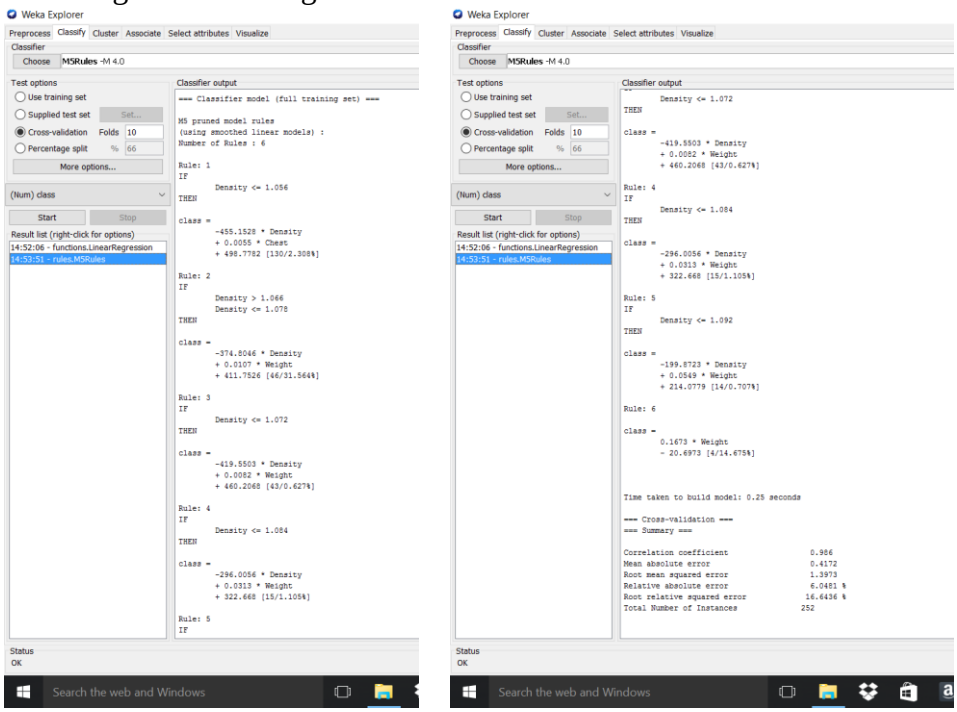
B. Classify/functions/LinearRegression was selected. The “test option”, “Cross-validation Fold 10” was selected. Further default values were maintained, and the following was generated:



Orysya Stus

The generated equation assigns different weights to the attribute values in combination in order to determine the value of the class, this equation has a correlation coefficient of 0.9862.
class = -410.2167 * Density + 0.0124 * Age + 0.0253 * Chest + 0.0314 * Abdomen + 446.1513

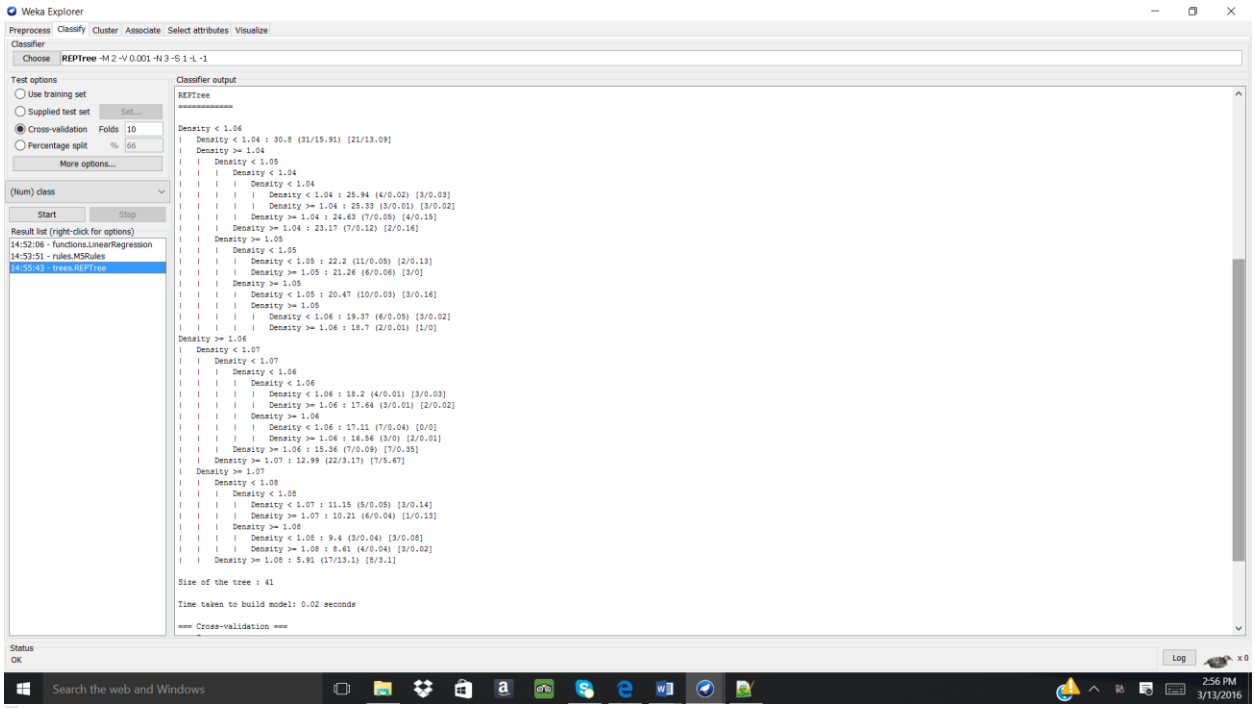
C. Classify/rules/M5Rules was selected with “Cross-validation Fold 10” as the test option. The following 6 rules were generated:

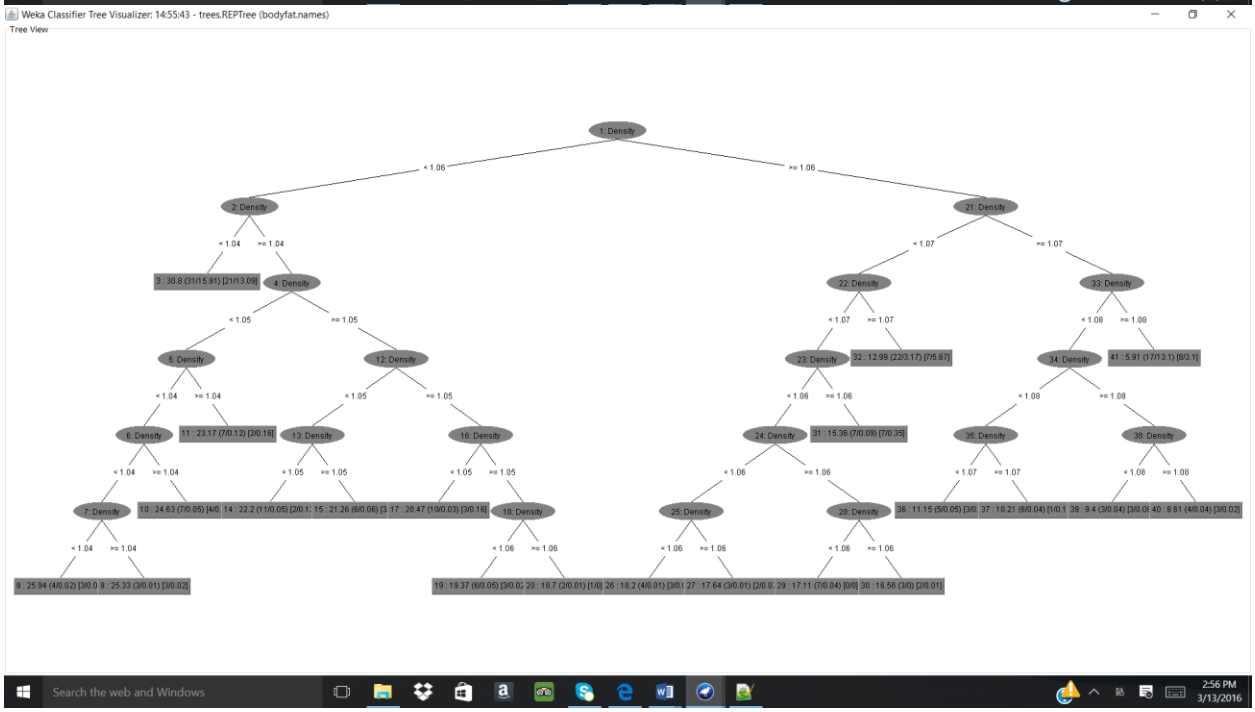


These rules are generated using “perfect” rules that use combinations of attribute values in order to find the value of “class”. The correlation coefficient is slightly lower compared to that of the linear regression equation, at 0.986.

D. Classify/trees/REPTree was selected with “Cross-validation Fold 10” as the test option.

Orysya Stus

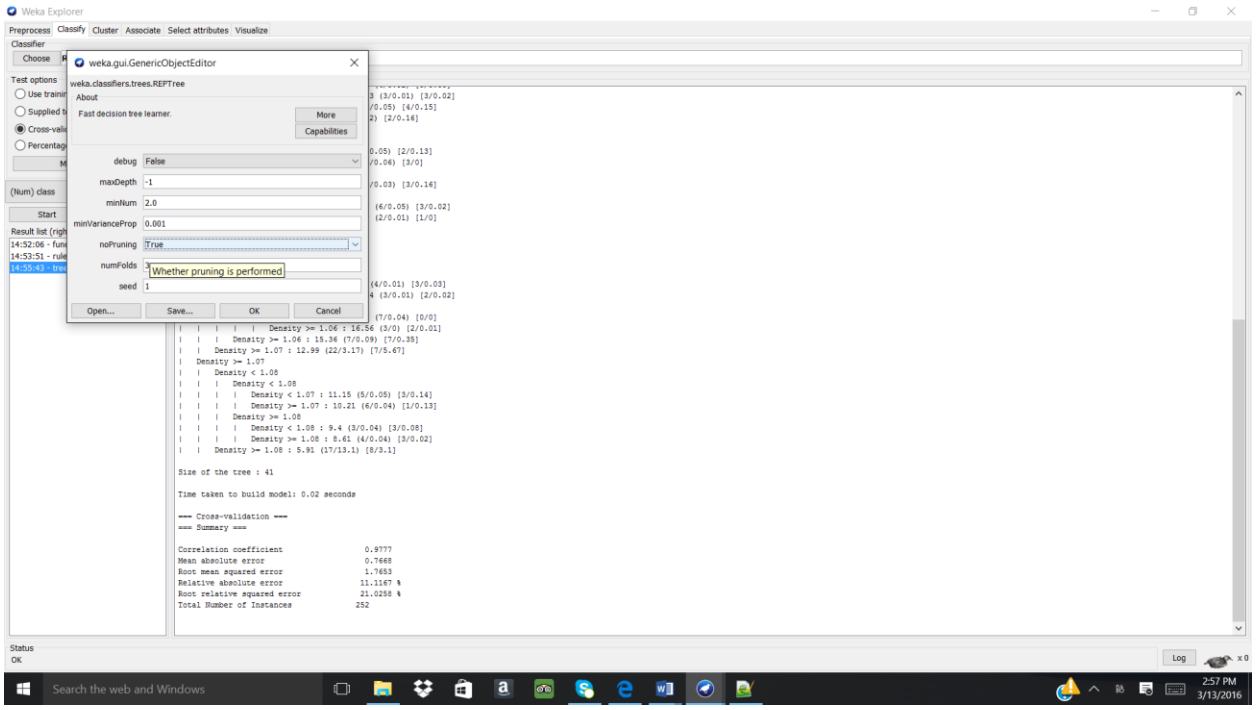




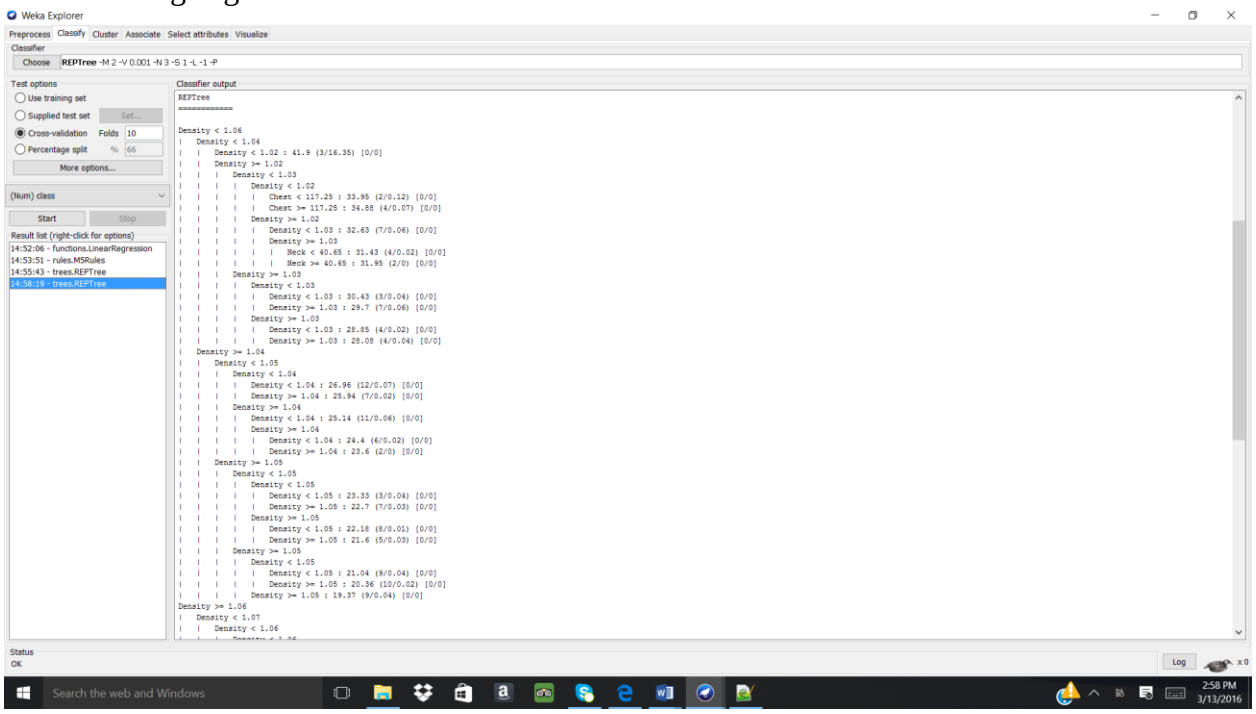
The tree generated has a lower correlation coefficient compared to the above learning schemes, at 0.9777. The model tree has many nodes.

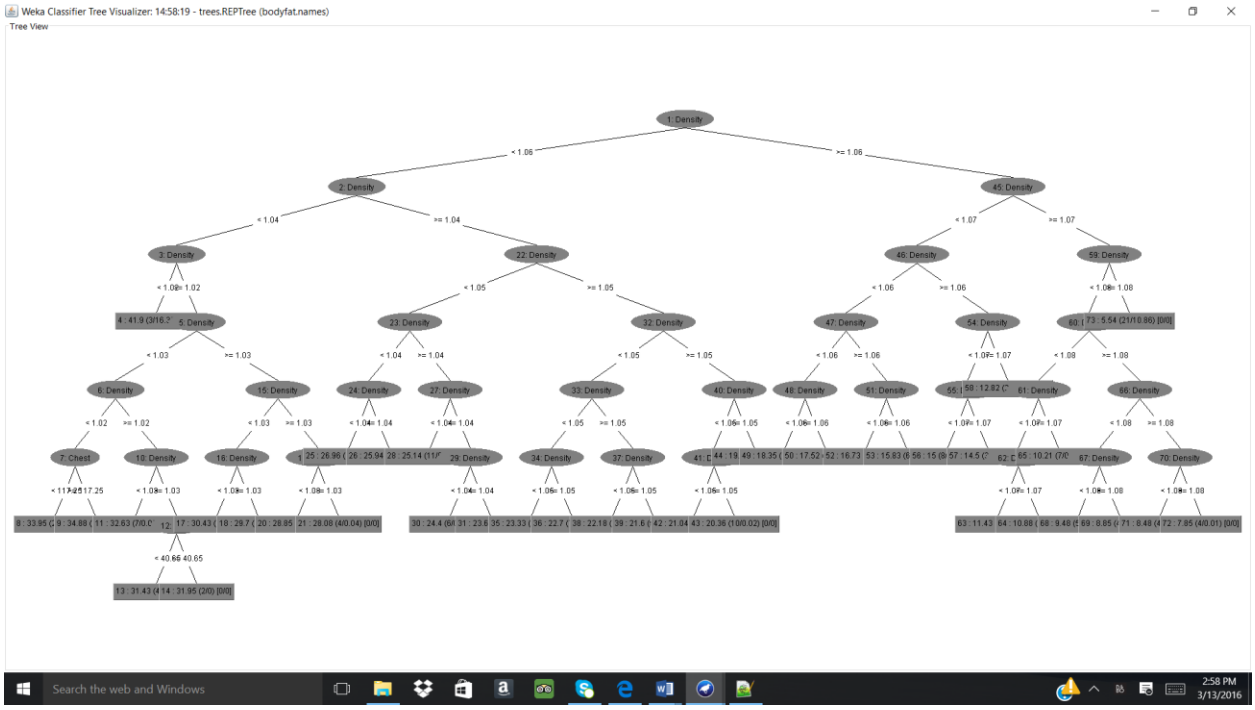
When the pruning factor is changed where “nopruning” is changed from false to true:

Orysya Stus



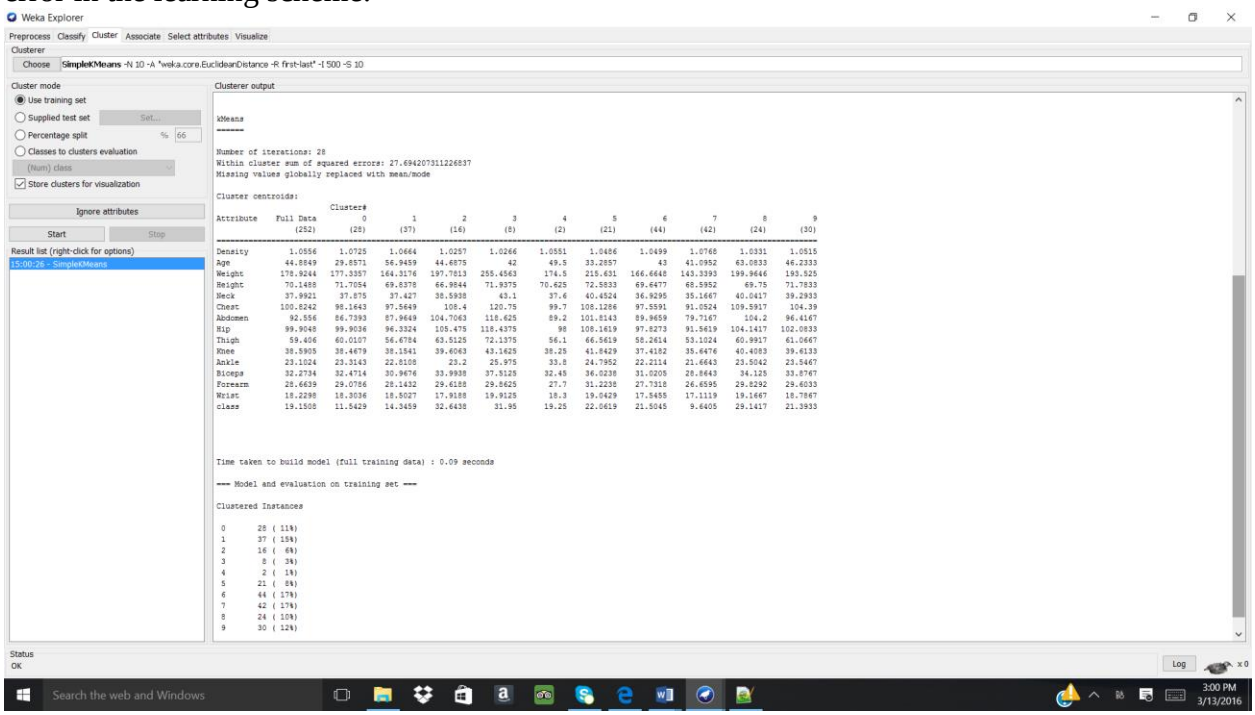
The following is generated:





The correlation coefficient increases to 0.9826 and the number of leaves greatly increases.

E. Cluster/clusterers/SimplekMeans was selected with the entire training set being used to generate the model. K = 10 was used for the number of clusters since this provided the lowest error in the learning scheme.

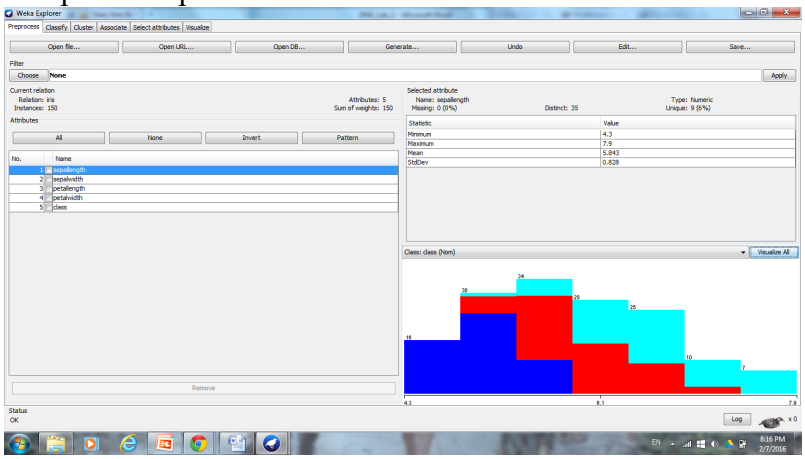


Taken together for this dataset, the highest correlation coefficient belonged to linear regression model and the M5 rules, showing that these two models would be the best to use in order to

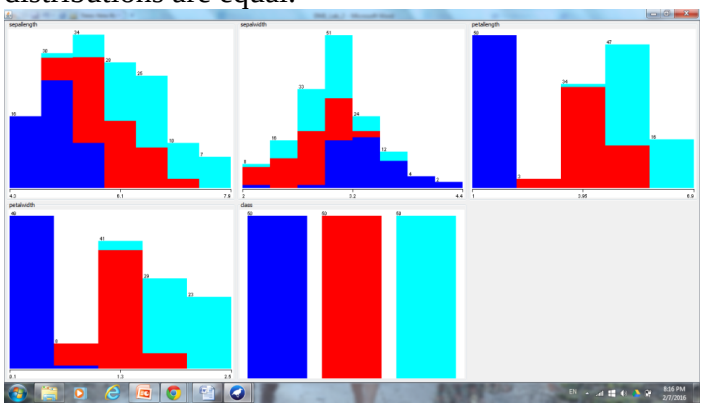
predict other BMIs. The attribute “Density” was provided the most information gain to this dataset in classifying the “class” of the BMI.

3. Use a k-means clustering technique to analyze the iris data set. What did you set the ***k*** value to be? Try several different values. What was the random seed value? Experiment with different random seed values. How did changing of these values influence the produced models?

A. The iris.arff was uploaded into Weka, there was no need to further filter the data during the “Preprocess” phase. Note: 5 attributes exist in this data set.

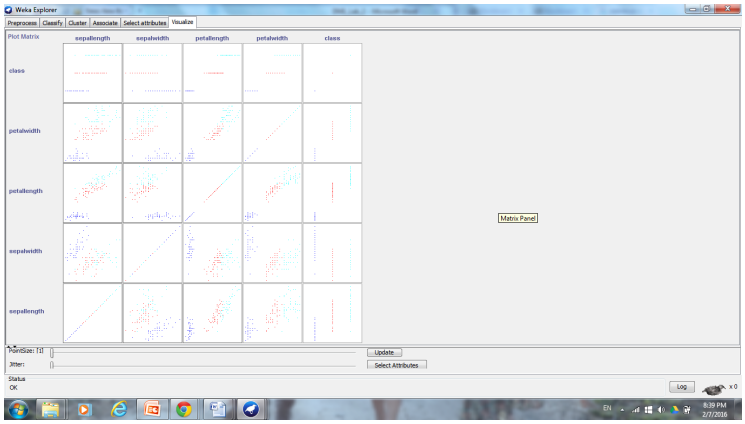


B. “Visualize all” was used to visual the distributions for each class. Where, attributes “sepalength” and “sepalwidth” follow positive skewed Gaussian distributions, the attributes “petallength” and “petalwidth” follow no concrete distribution, and the “class” attribute distributions are equal.

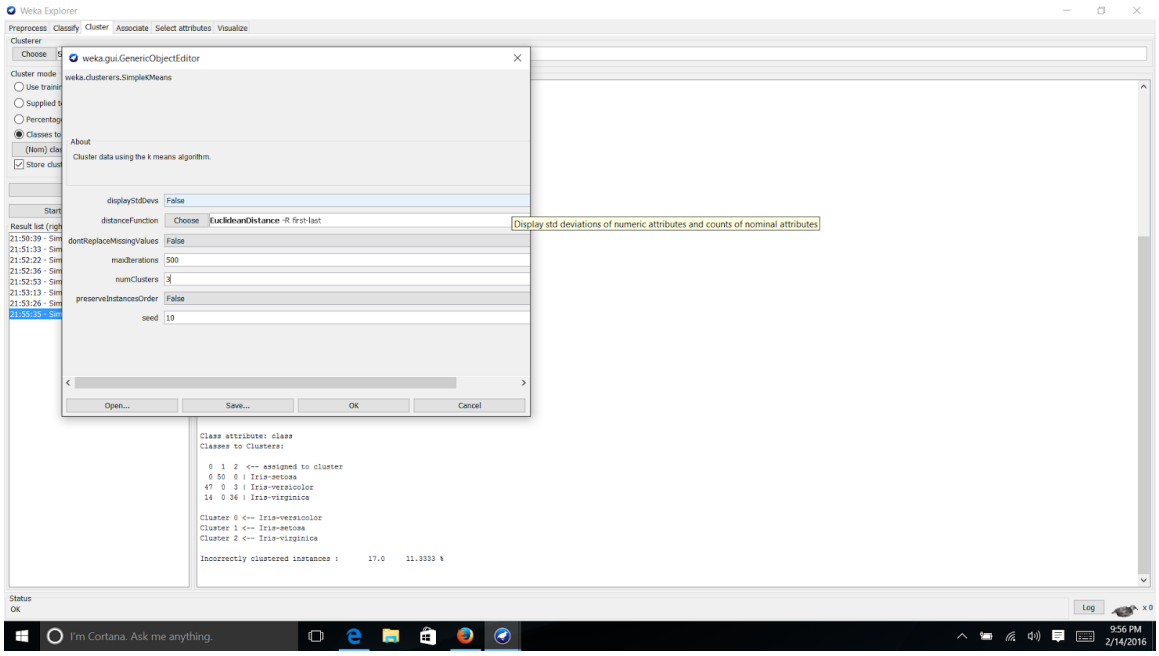


C. Utilizing the “Visualize” tab, clusters for attribute values can be seen with the most apparent clusters seen for petalwidth vs. sepalwidth/petallength and petallength vs. sepallength/sepalwidth/petalwidth. These attributes will be utilized in order to determine the class (Iris-setosa, Iris-versicolor, or Iris-virginica).

Orysya Stus

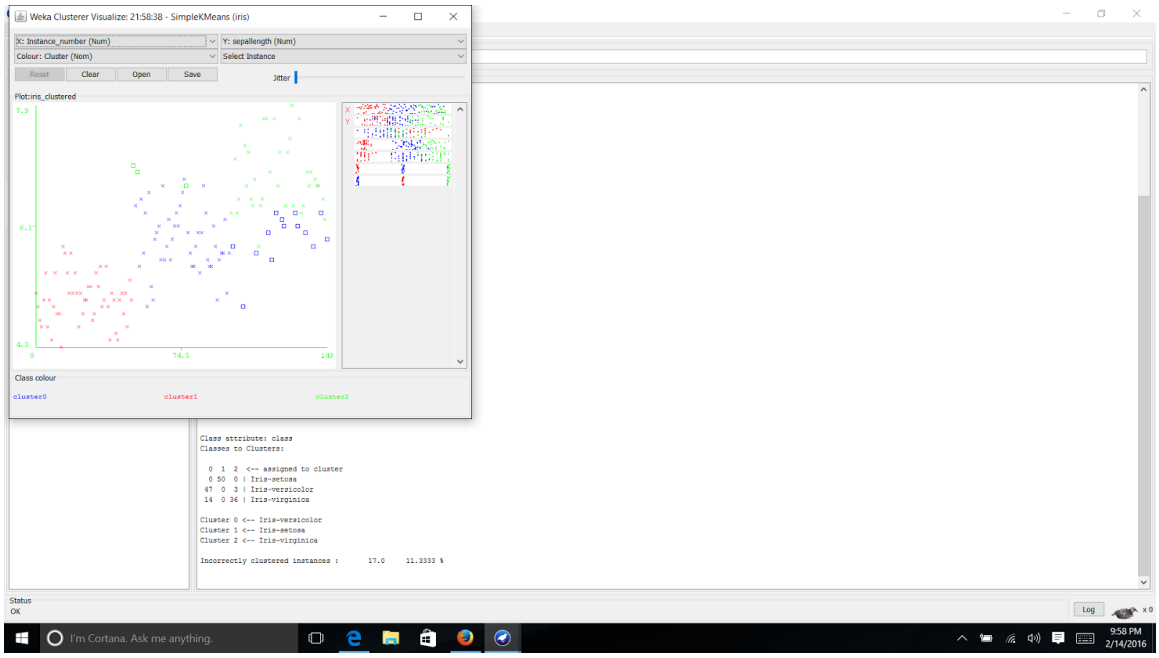


D. To apply the k-means clustering technique, under the “Cluster” tab Clusterers/SimpleKMeans is selected. At the pop-up window, “numClusters” was changed to 3 since the number of class attributes is 3, “seed” was maintained at 10.



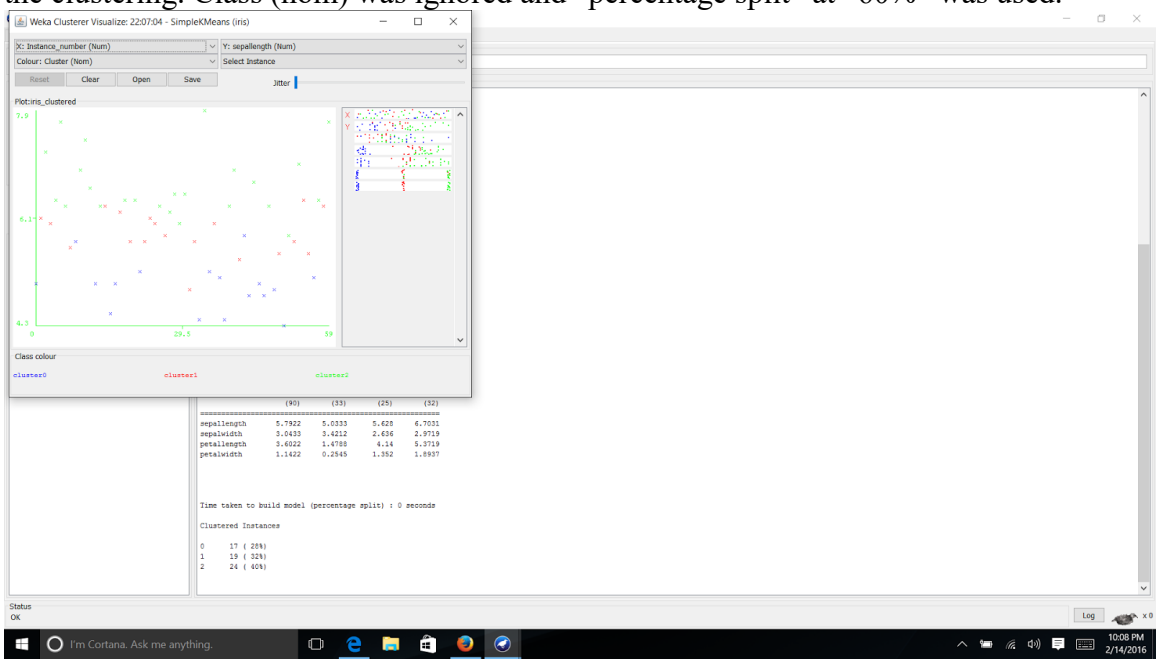
Note: for the first means of analysis, the class attribute was not ignored and the “classes to cluster evaluation” was set to “class (nom)”.

Orysya Stus



Three clusters were created with the “Clustered Instances” containing 61 (41%) in cluster 0, 50 (33%) in cluster 1, and 39 (26%) in cluster 2. Note: in the class distribution in Part B, Iris-setosa, Iris-versicolor, and Iris-virginica had equal distributions.

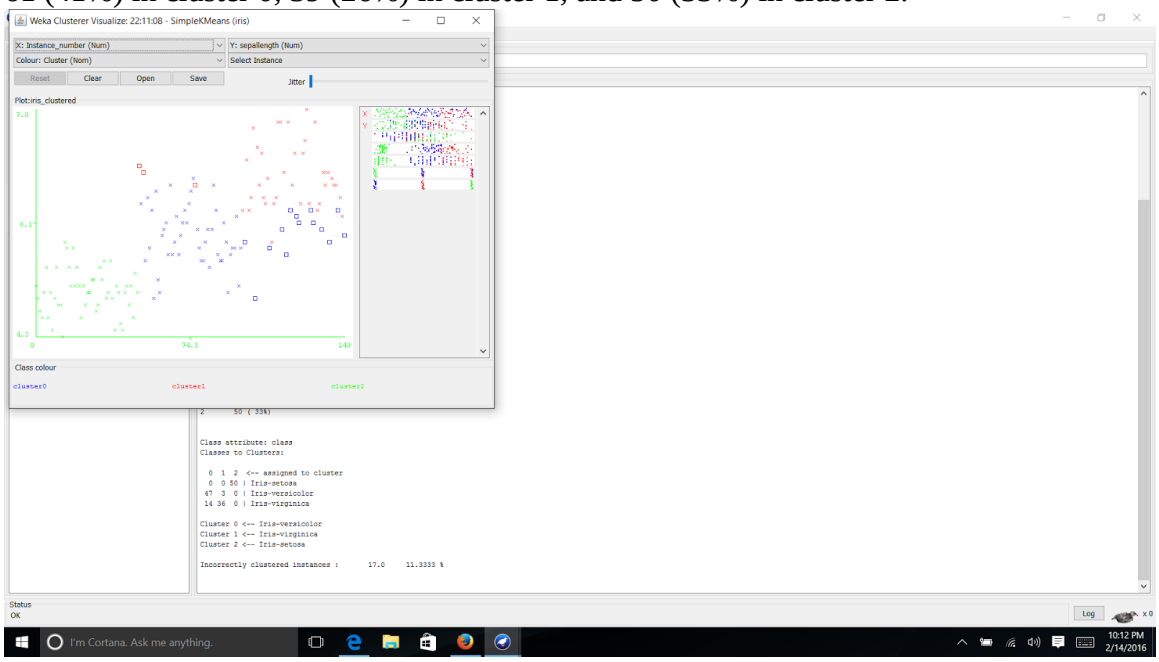
But as clustering is unsupervised, next the class attribute was ignored and only the four attributes - sepallength, sepalwidth, petallength, and petalwidth were considering in clustering to compare the clustering. Class (nom) was ignored and “percentage split” at “60%” was used.



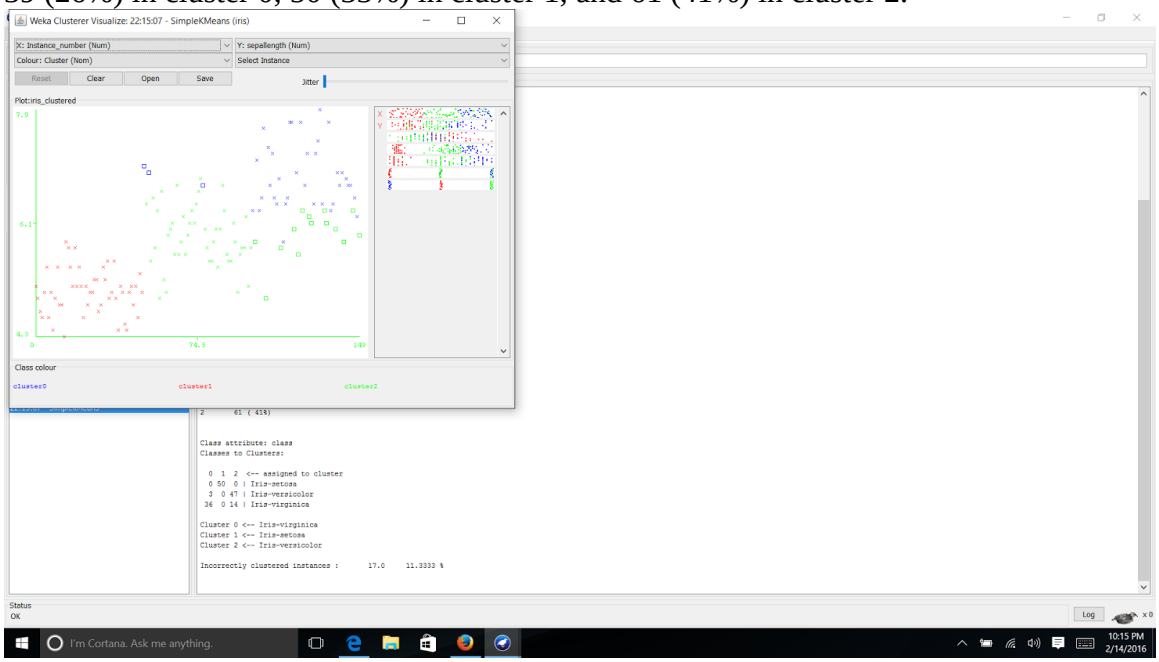
Making the above adjustments, changed the clustering vastly in comparison with the cluster centroids being located in different places then before.

Furthermore, parameter variations for the first cluster produced, where “class (nom)” is not ignored were made.

When increasing the “seed” from 10 to 50, the clusters looked similar to that shown above with 61 (41%) in cluster 0, 39 (26%) in cluster 1, and 50 (33%) in cluster 2.



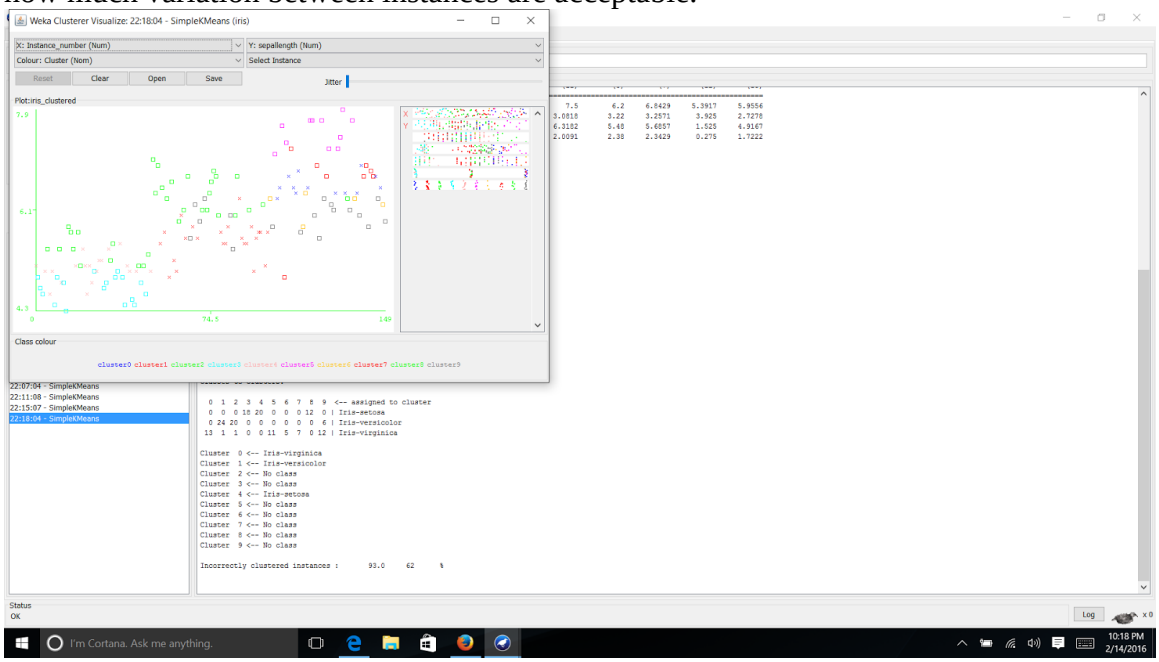
When decreasing the “seed” from 10 to 5, the clusters looked similar to that shown above with 39 (26%) in cluster 0, 50 (33%) in cluster 1, and 61 (41%) in cluster 2.



Note: It is interesting that the ratio 39:50:61 is conserved when changing the “seed” number. The seed number (any integer) is the randomization for your initial K points. K represents the number of clusters. Because K-means is sensitive to initial points, you will have to try experimentation

on the stability of your clusters with different seeds. However, K is user defined which could also be guided by domain knowledge and other practical factors.

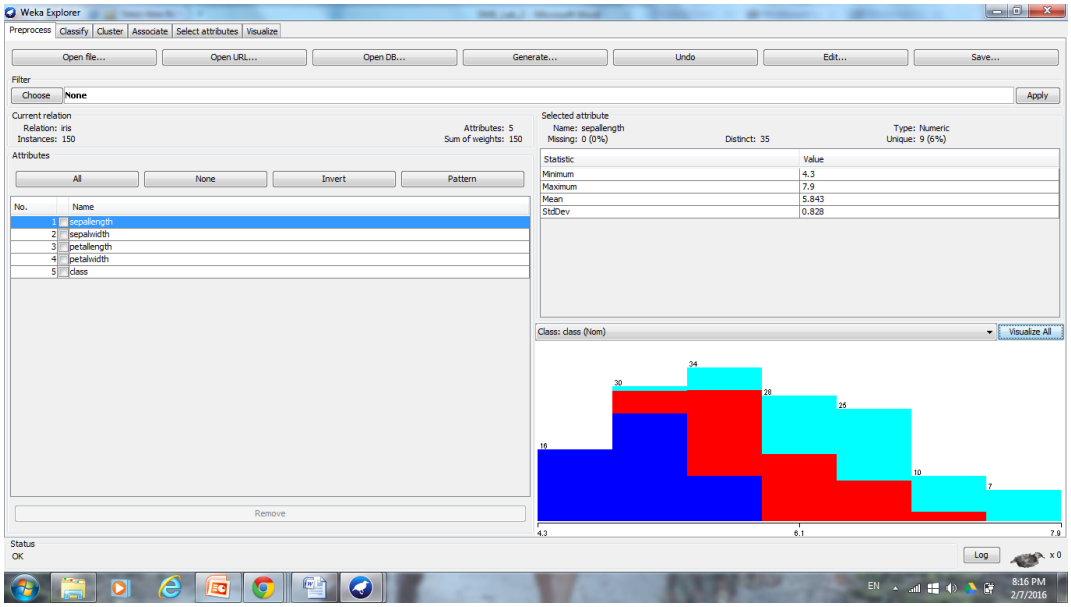
Changing the number of clusters from 3 to 10, does not create nice clean clusters for all. Nicer clusters can be created after recursive measures of visualizing the clustering and determining how much variation between instances are acceptable.



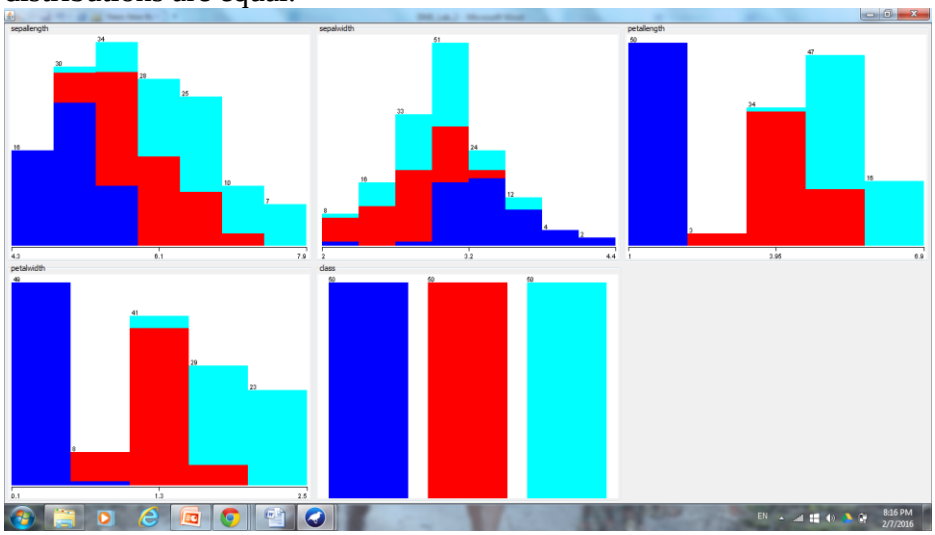
4. Produce a hierarchical clustering (COBWEB) model for iris data. How many clusters did it produce? Why? Does it make sense? What did you expect?
   Change the acuity and cutoff parameters in order to produce a model similar to the one obtained in the book. Use the classes to cluster evaluation – what does that tell you?

A. The iris.arff was uploaded into Weka, there was no need to further filter the data during the "Preprocess" phase. Note: 5 attributes exist in this data set.

Orysya Stus

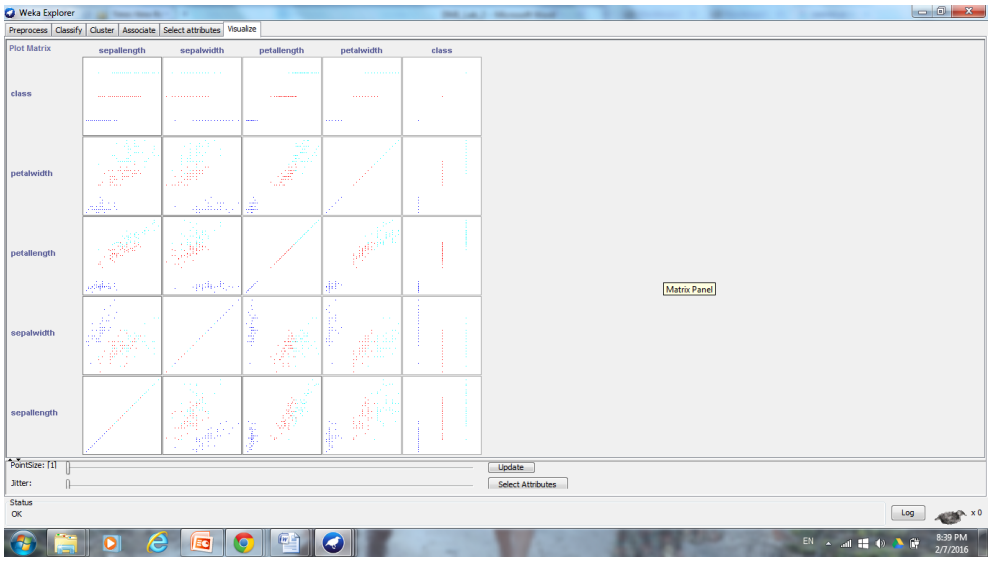


B. “Visualize all” was used to visual the distributions for each class. Where, attributes “sepalength” and “sepalwidth” follow positive skewed Gaussian distributions, the attributes “petallength” and “petalwidth” follow no concrete distribution, and the “class” attribute distributions are equal.



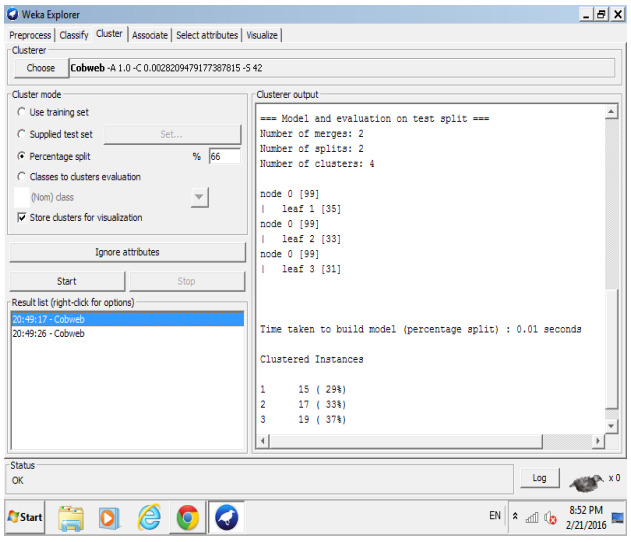
C. Utilizing the “Visualize” tab, clusters for attribute values can be seen with the most apparent clusters seen for petalwidth vs. sepalwidth/petallength and petallength vs. sepallength/sepalwidth/petalwidth. These attributes will be utilized in order to determine the class (Iris-setosa, Iris-versicolor, or Iris-virginica).

Orysya Stus

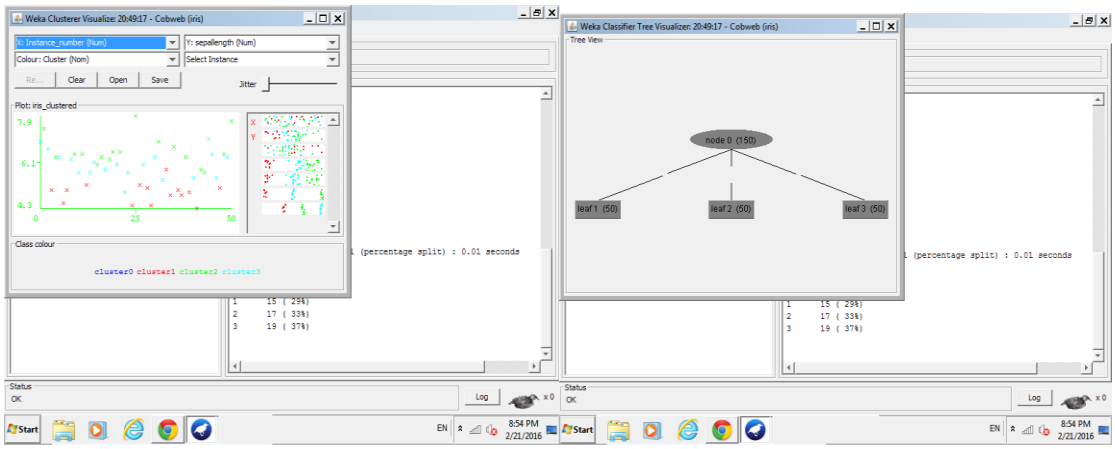


D. Under Cluster/COBWEB, the “Percentage split” at 66% was used to test the data set. COBWEB, utilizes incremental clustering on nominal values. All of the values in the iris data set are nominal. The analysis was run first with the “class” attribute as well as ignoring the attribute “class”:

With the “class” attribute:

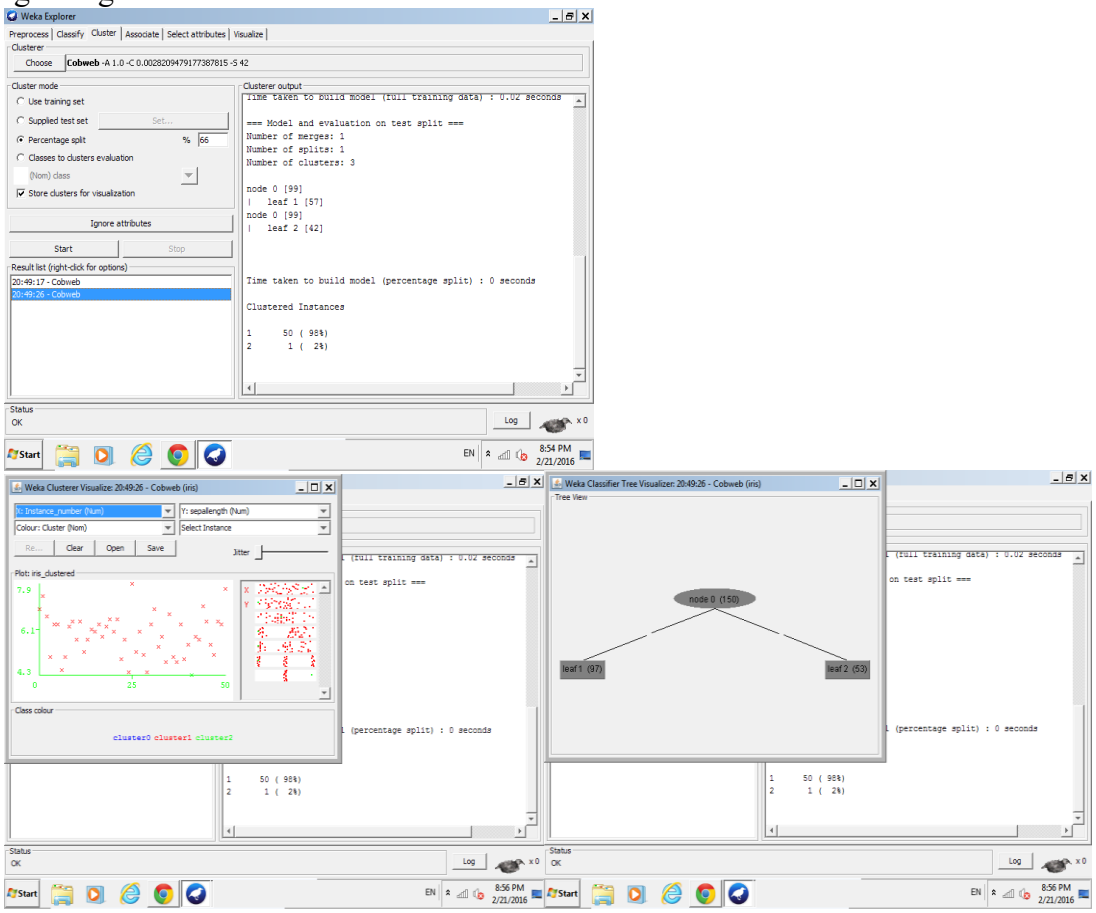


Orysya Stus



There are three leafs/clusters produced, which makes sense since the class attribute has 3 different values (Iris-setosa, Iris-versicolor, or Iris-virginica at 50:50:50).

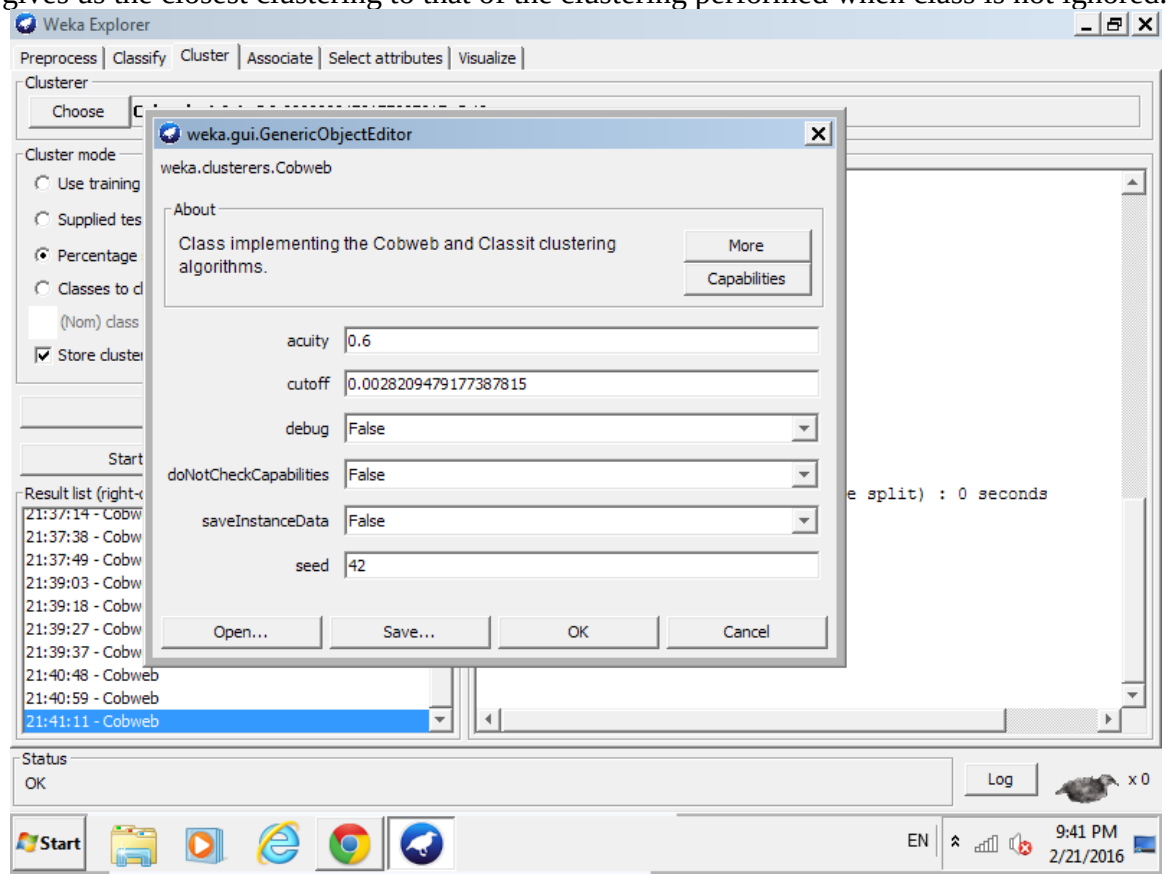
Ignoring the "class" attribute:

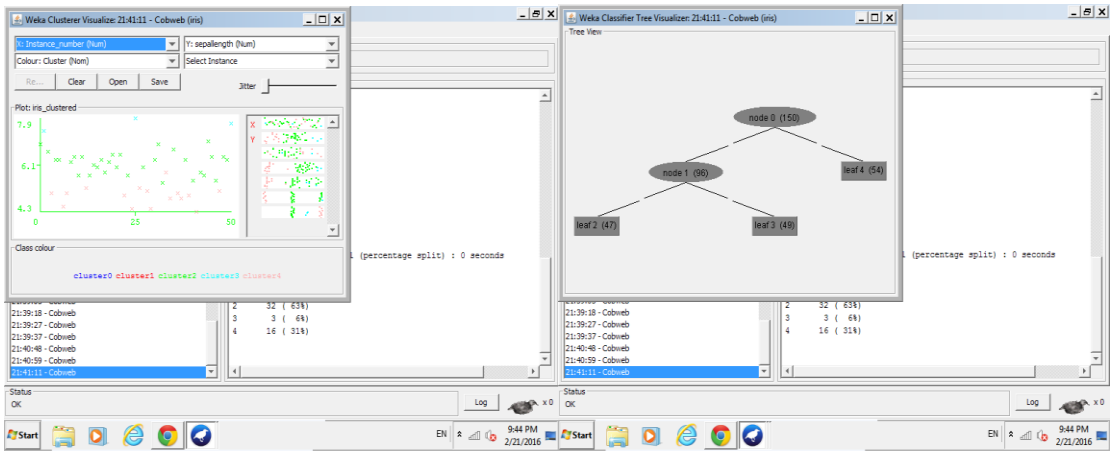


When ignoring the class attribute, only 2 cluster/leafs are generated with a 97:53 ratio which does not clearly cluster the data as well as the clustering above.

Ignoring the class, further changes to the acuity and cutoff parameters will be made to try to cluster the Iris types together and get a result as similar as possible to when the class is not ignored. Through systematic experimentation with acuity and cutoff parameters, it was found

that utilizing a acuity of 0.6 and maintaining the cutoff at the default 0.0028209479177387815, gives us the closest clustering to that of the clustering performed when class is not ignored.





The ratio of the clusters/leaves are 47:49:54, which is the closest to the 50:50:50 ration seen previously.