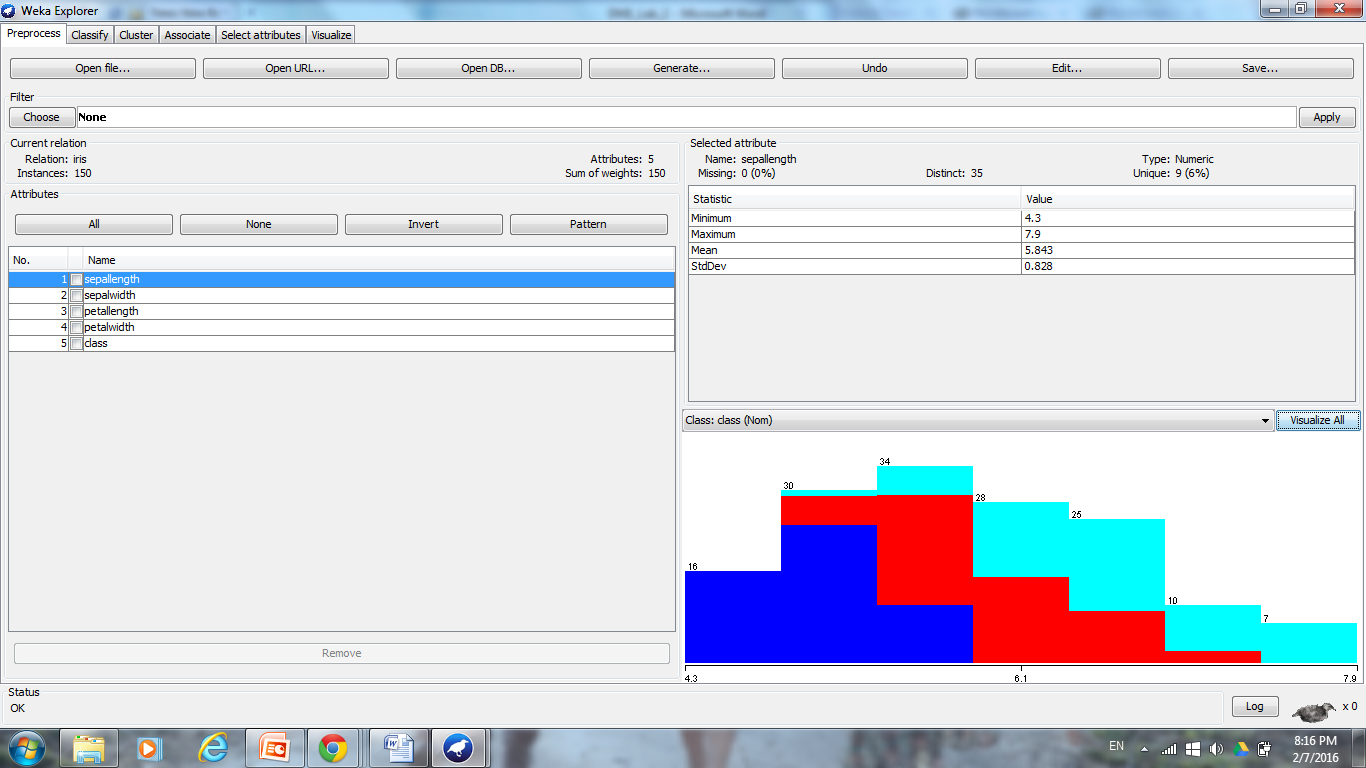
### **Data Mining I: Basic Methods and Techniques**

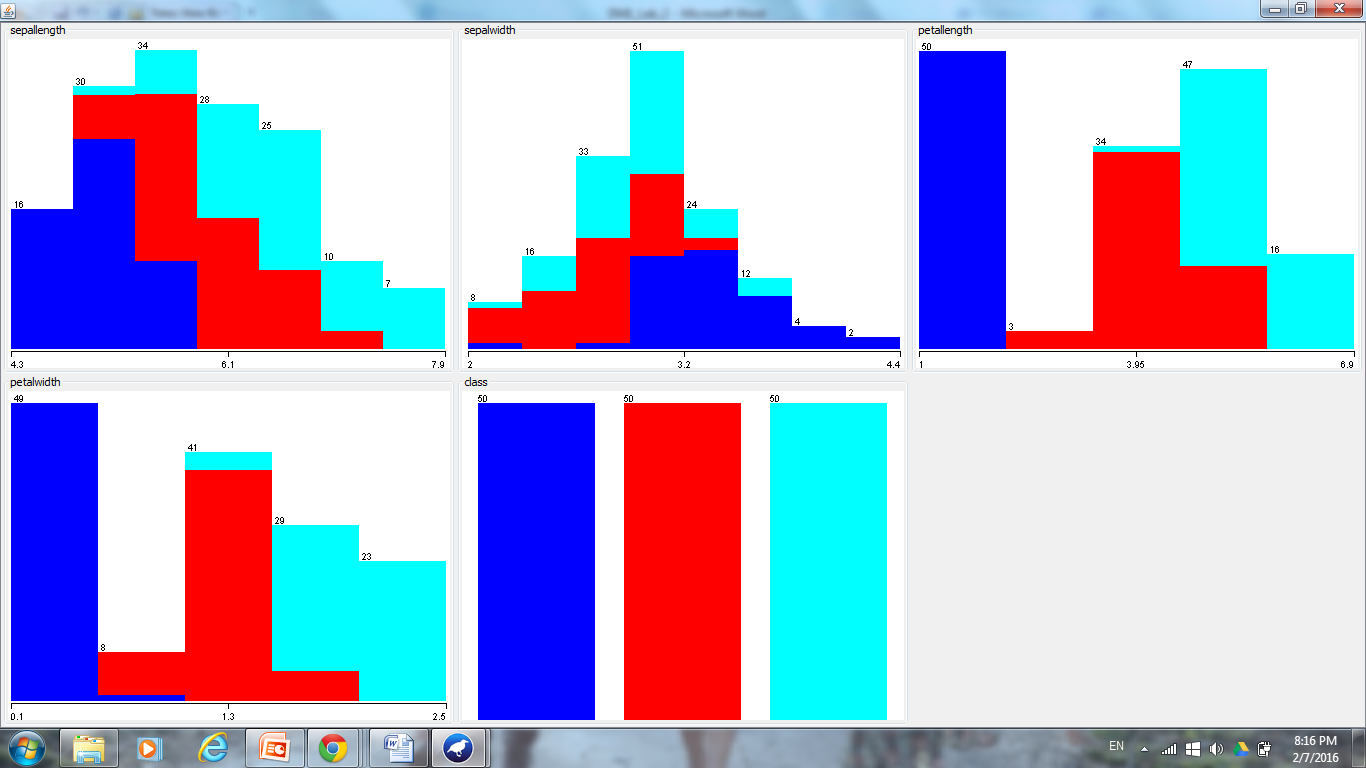
***Laboratory Assignment #2:***

The following was done:

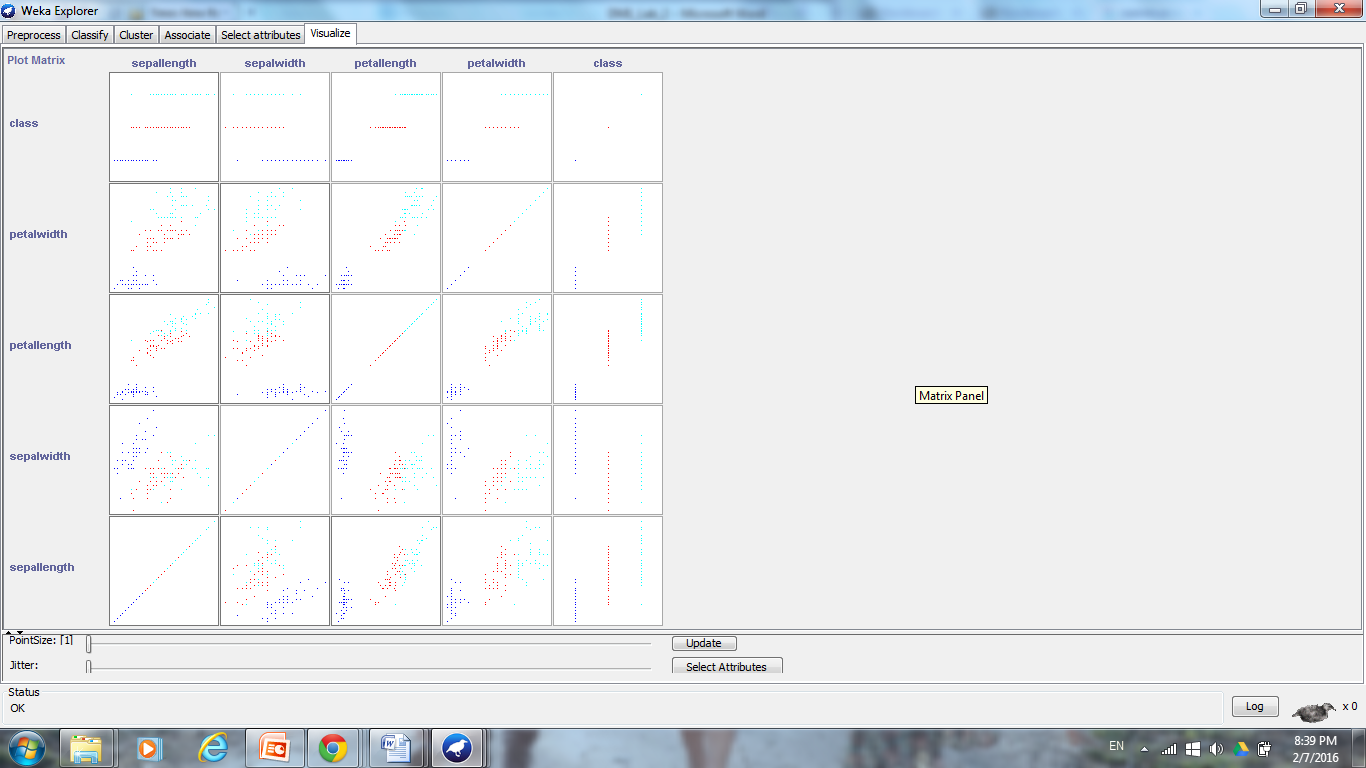
A. The iris.arff was uploaded into Weka, there was no need to further filter the data during the “Preprocess” phase. Note: 5 attributes exist in this data set.



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).

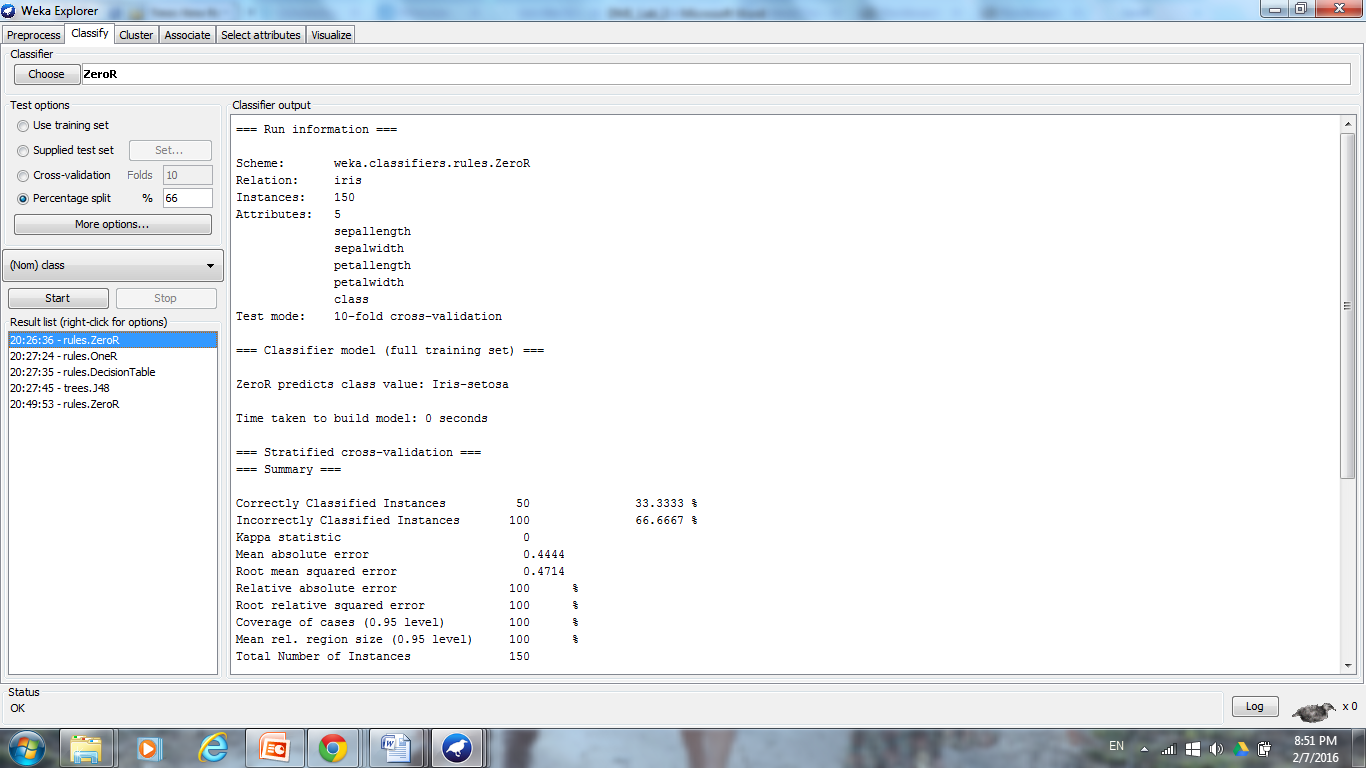


D. Clicking on the “Classify” tab, 3 different types of rules (ZeroR, OneR, and Decision table) and 1 tree (J48) completed. Under “Test options”, “Cross-validation” at “Fold 10” was selected in order to accomplish the most believable evaluations where the data is split into training and test data sets 10 times with models being created, iterated, and polished between each training and testing.

Note: Individual discussion is completed below with screenshots for referencing the data.

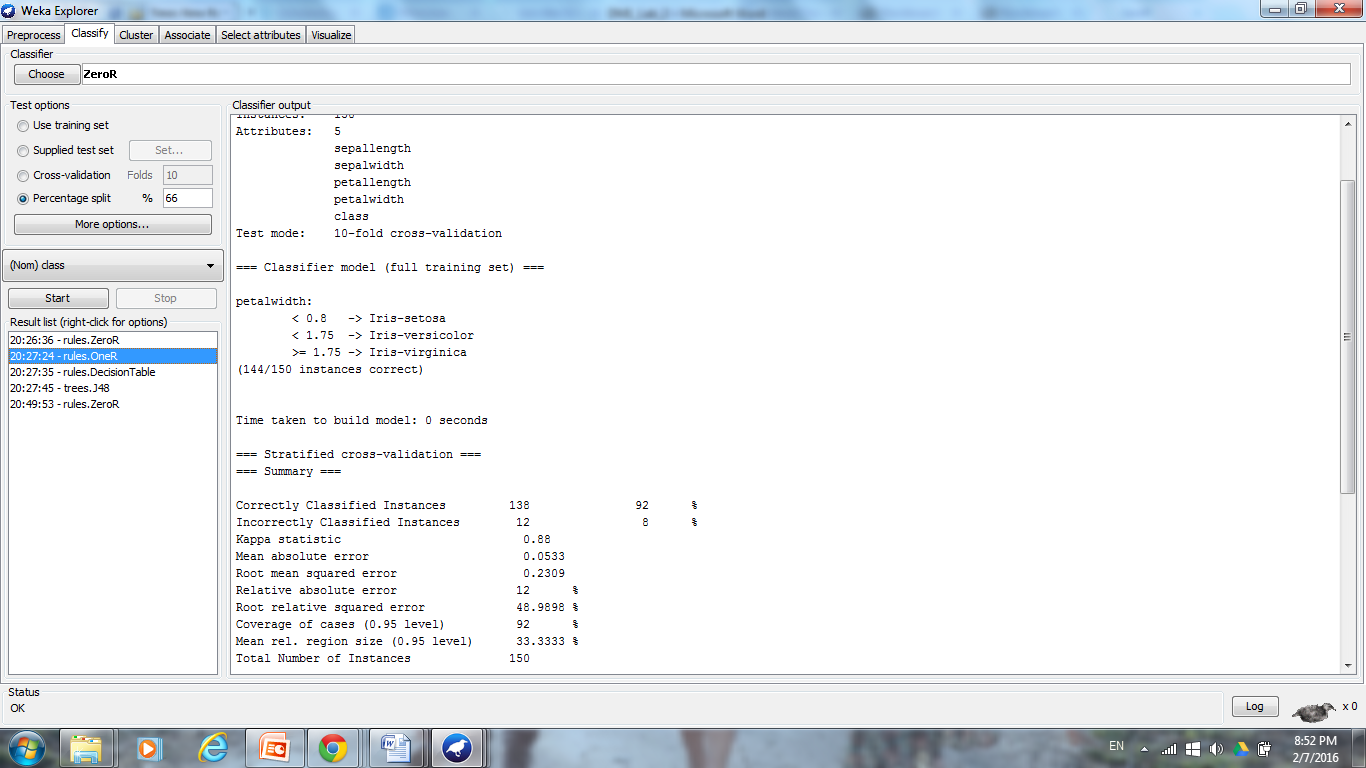
**ZeroR:**

This algorithm constructs a frequency table for the target and selects its most frequent value. In considering the “class” attribute distribution during the “Pre-process phase”, it should be noted that the classes: Iris-setosa, Iris-versicolor, or Iris-virginica are represented equally at a 1:1:1 ratio, yet this generated algorithm predicted Iris-setosa to be the most frequent. This is false and is associated with a high error rate. This algorithm was able to map 50 instances correctly, while 100 instances did not follow this rule; the mean absolute error was high at 0.4444. This classifier rule performed poorly on the iris.arff dataset and should not be used to model the data.



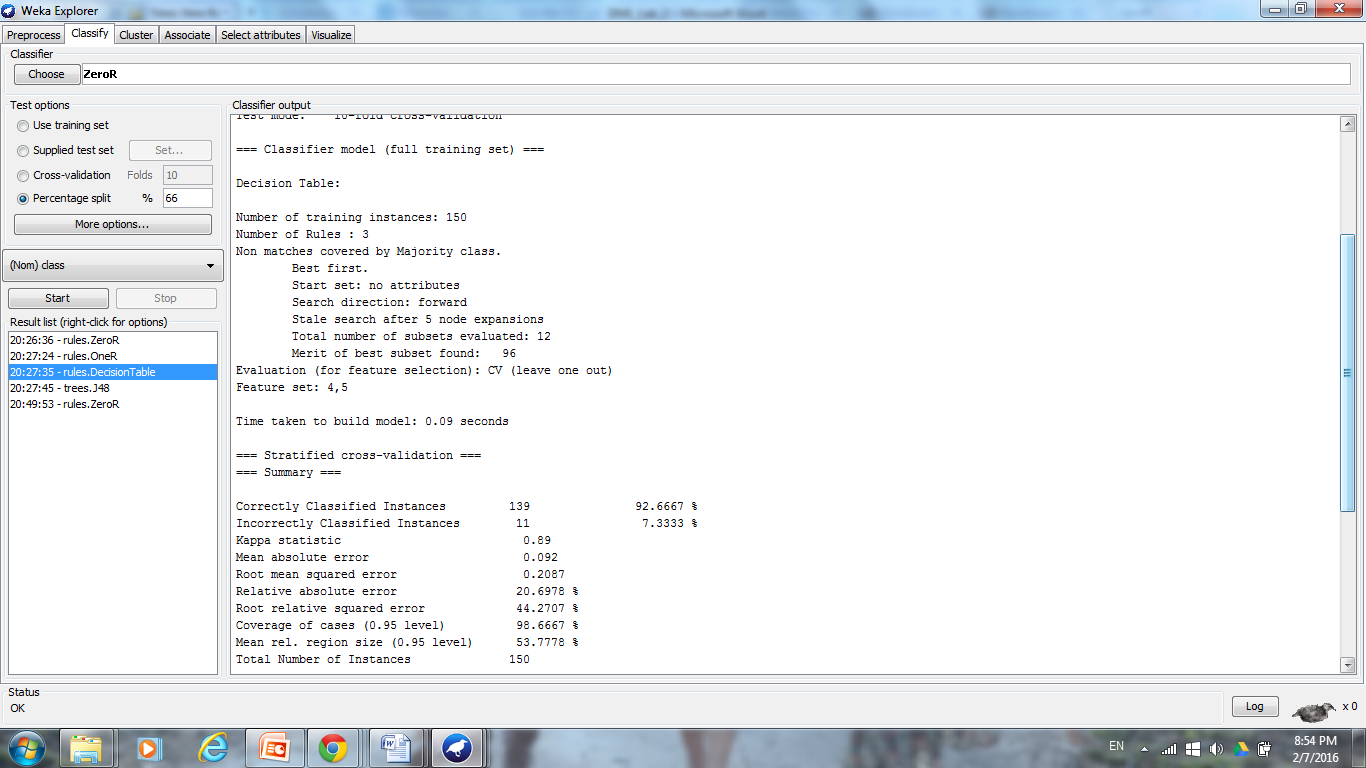
**OneR:**

The algorithm creates a model based on a single attribute, to determine the classes for the datasets. The attribute “petalwidth” 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 classes: Iris-setosa, Iris-versicolor, or Iris-virginica. This rule maps 138 instances correctly, but 12 incorrectly with a mean absolute error of 0.0533. 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.



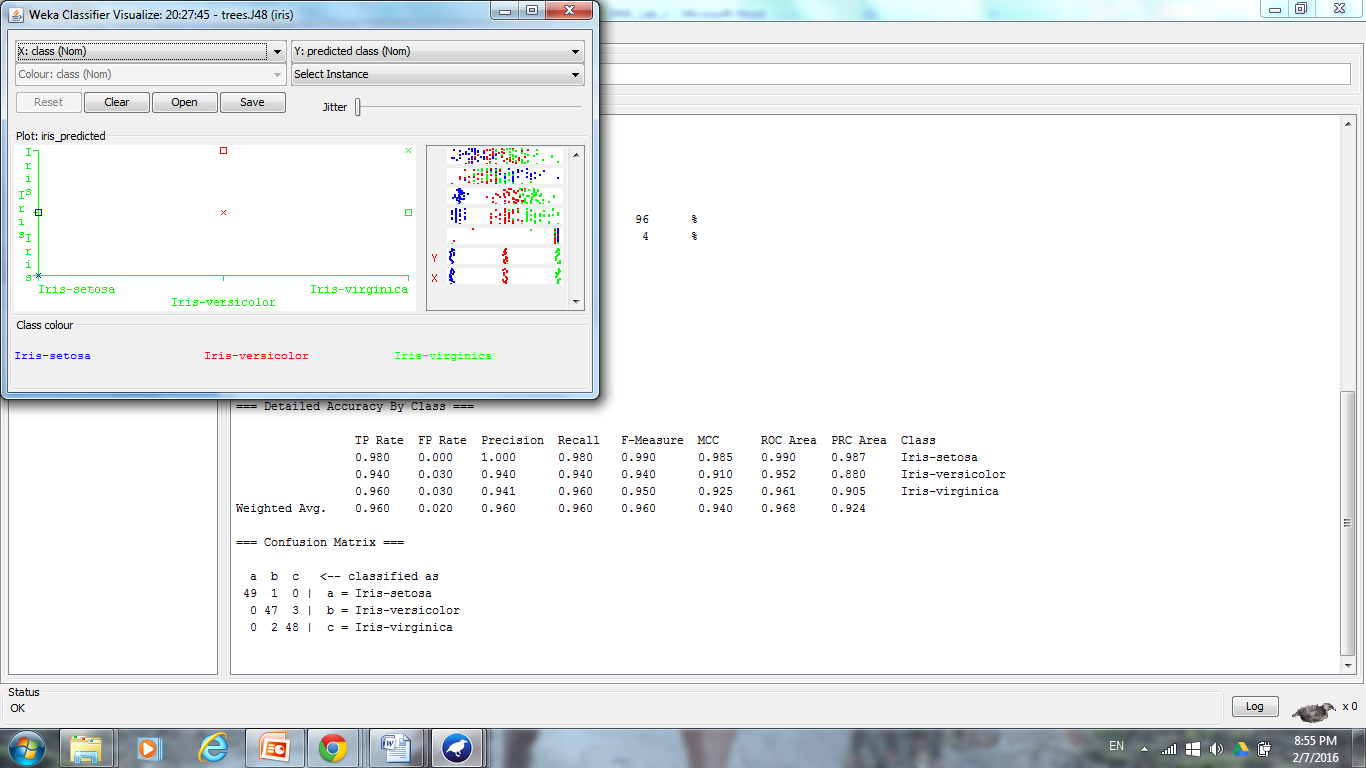
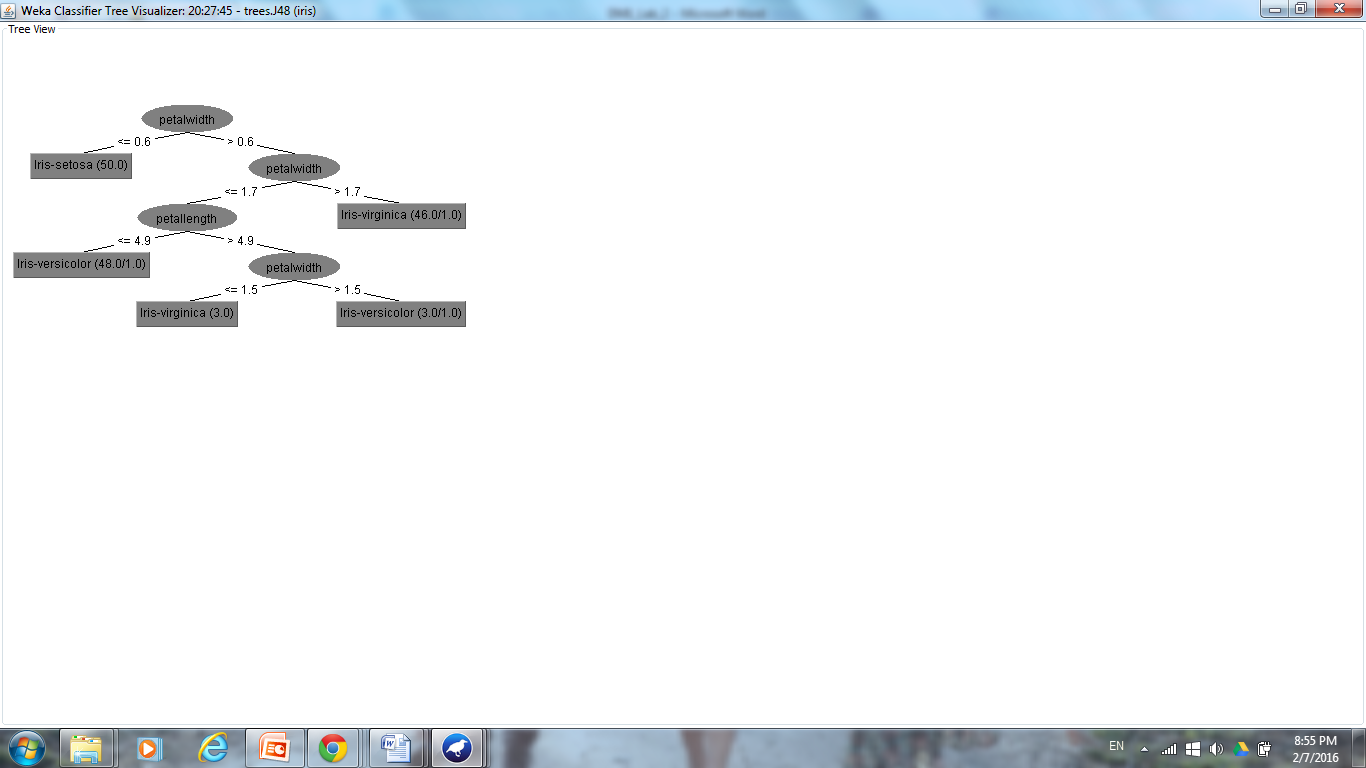
**Decision Table:**

The Decision table creates a table which includes the attributes making the decision or classifying instances into a class to be included. Through this algorithm, 3 decision rules were created in order to classify the attributes with 139 correctly classified, 11 incorrectly, and a mean absolute error of 0.092. The attributes which were featured are: “petalwidth” and “class” showing that these to attributes provide the most supervised learning possible in the data set. Although the decision table classified more instances correctly compared to the OneR rule, this algorithm has a higher error rate. Therefore, further test data may be necessary to determine whether a decision table rule or OneR rule can classify more instances. A more complex model, like a decision tree can provide more information and less error when classifying.



**Decision Tree:**

The J48 pruned tree creates a decision tree classifier model which creates the shortest tree which provides the most information; the background algorithm computes which attributes will provide the most gain and thus should be used as a node. This C4.5 classifier model classifies 144 instances correctly, 6 instances incorrectly, and has a mean absolute error of 0.035 147. Visualization of the decision tree shows that the attribute which provides the most information gain is “petalwidth”, which was confirmed as well by the previous rules. Further analysis of the decision tree shows that “petalwidth” is used at the nodes 3 times thus showing that more complex rules were necessary in order to classify the instances. Still, this model provides the most amount of information for the iris.arff dataset, has the lowest error rate making it the most reliable, and overfitting is avoided.

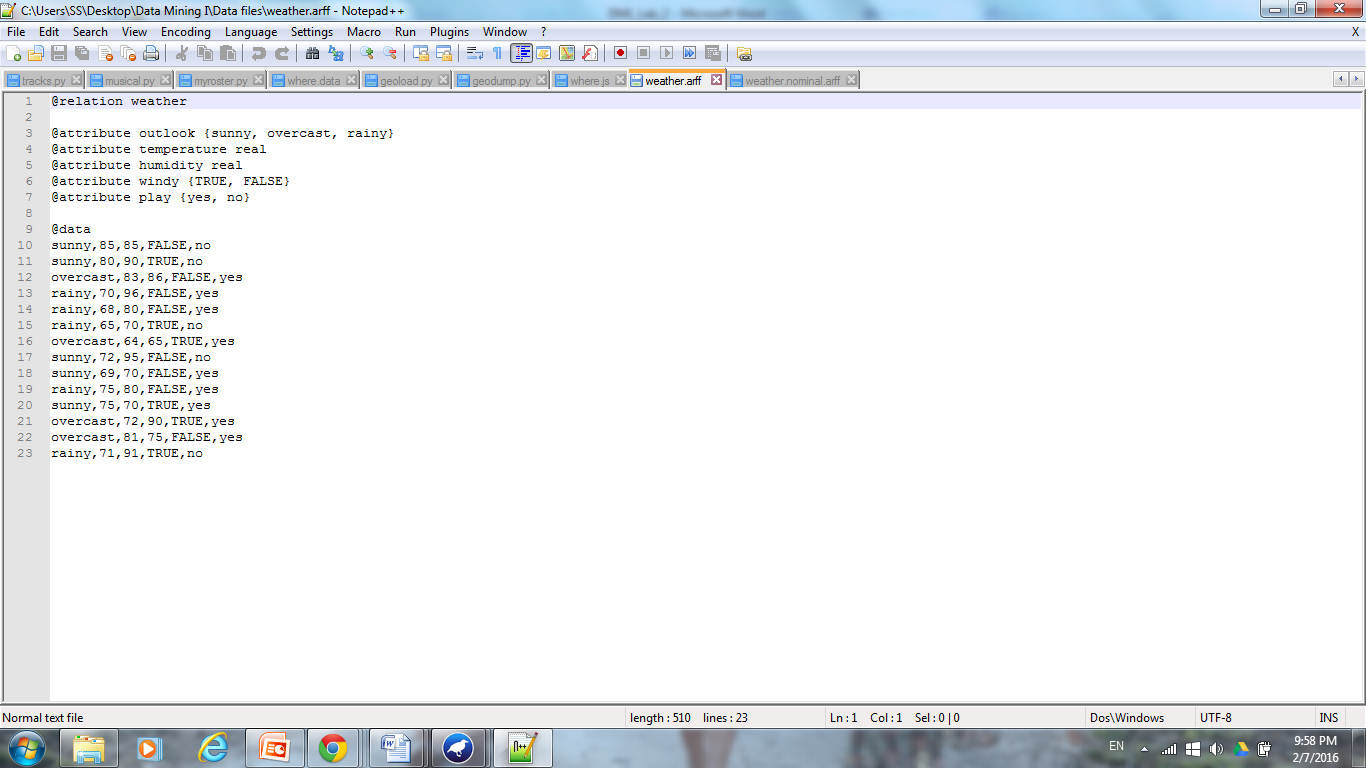


2.

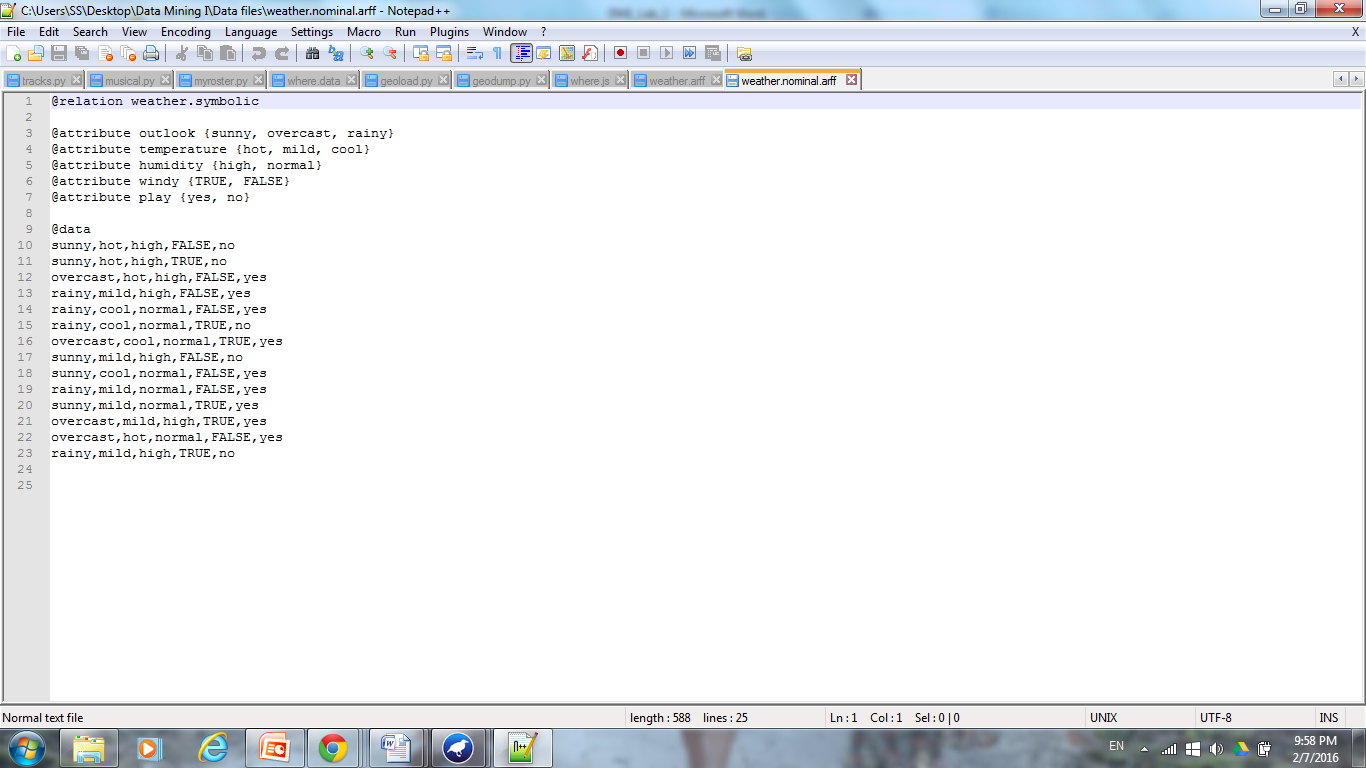
The following was done:

A. The weather.arff and weather\_nominal.arff datasets were compared via Notepad in order to determine how the attribute values correlated to each object of the instance. Utilizing only nominal attribute values in weather\_nominal.arff, instead of the weather.arff real numeric values will allow for easier visualization and better classification.

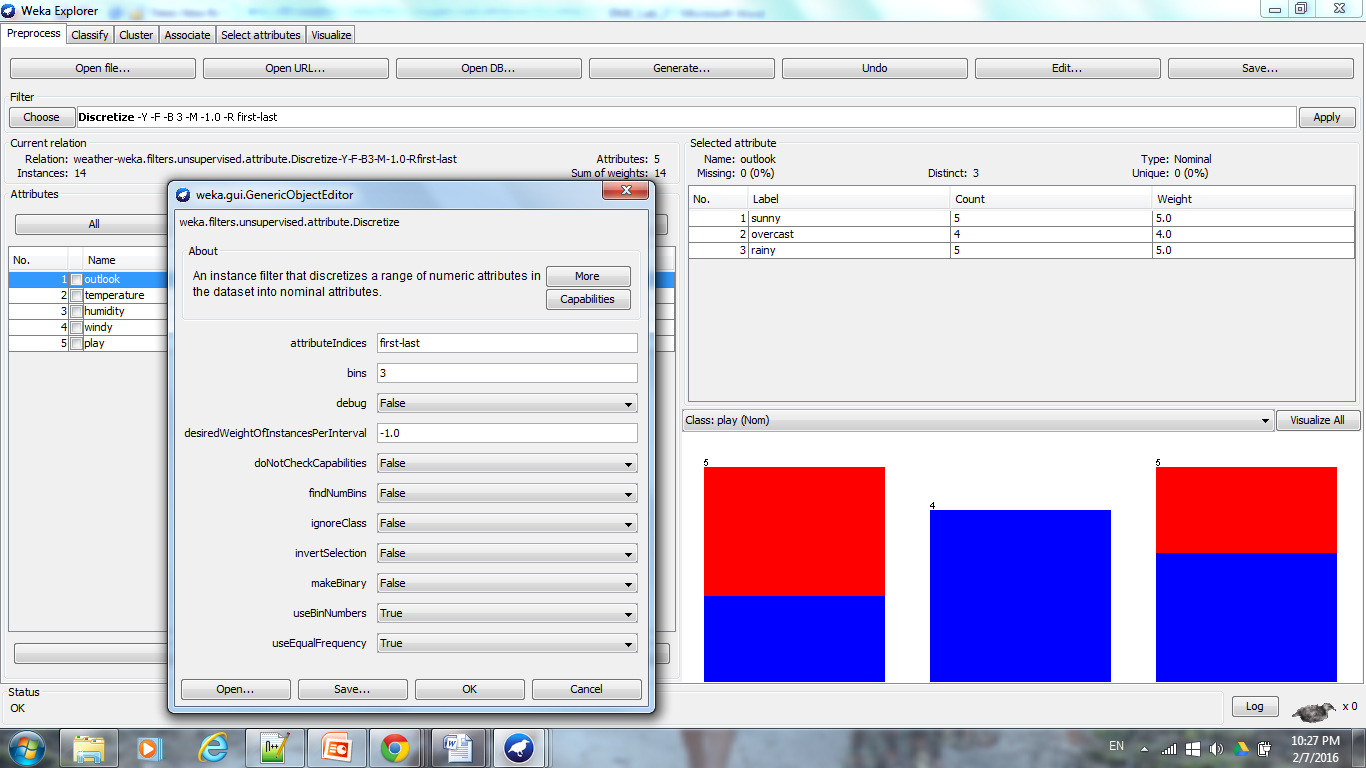
weather.arff:



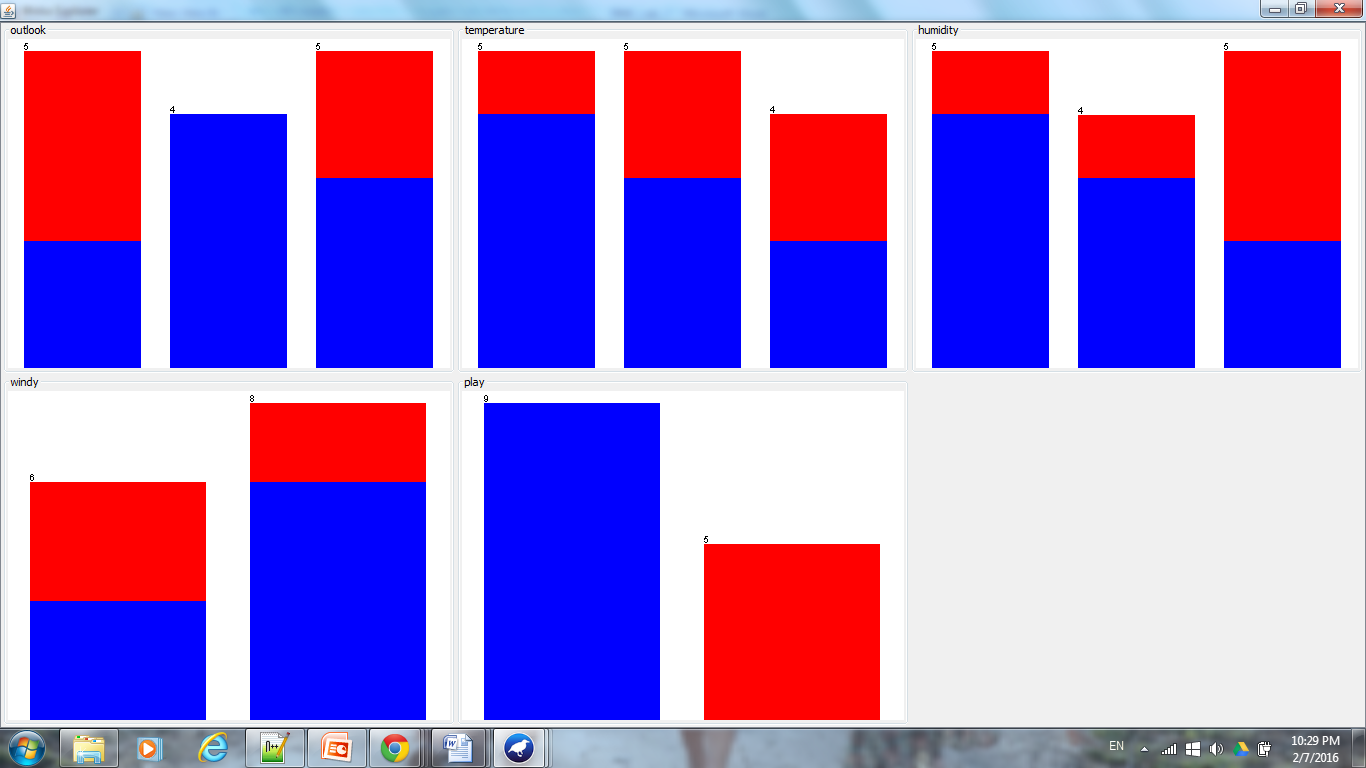
weather\_nominal.arff:



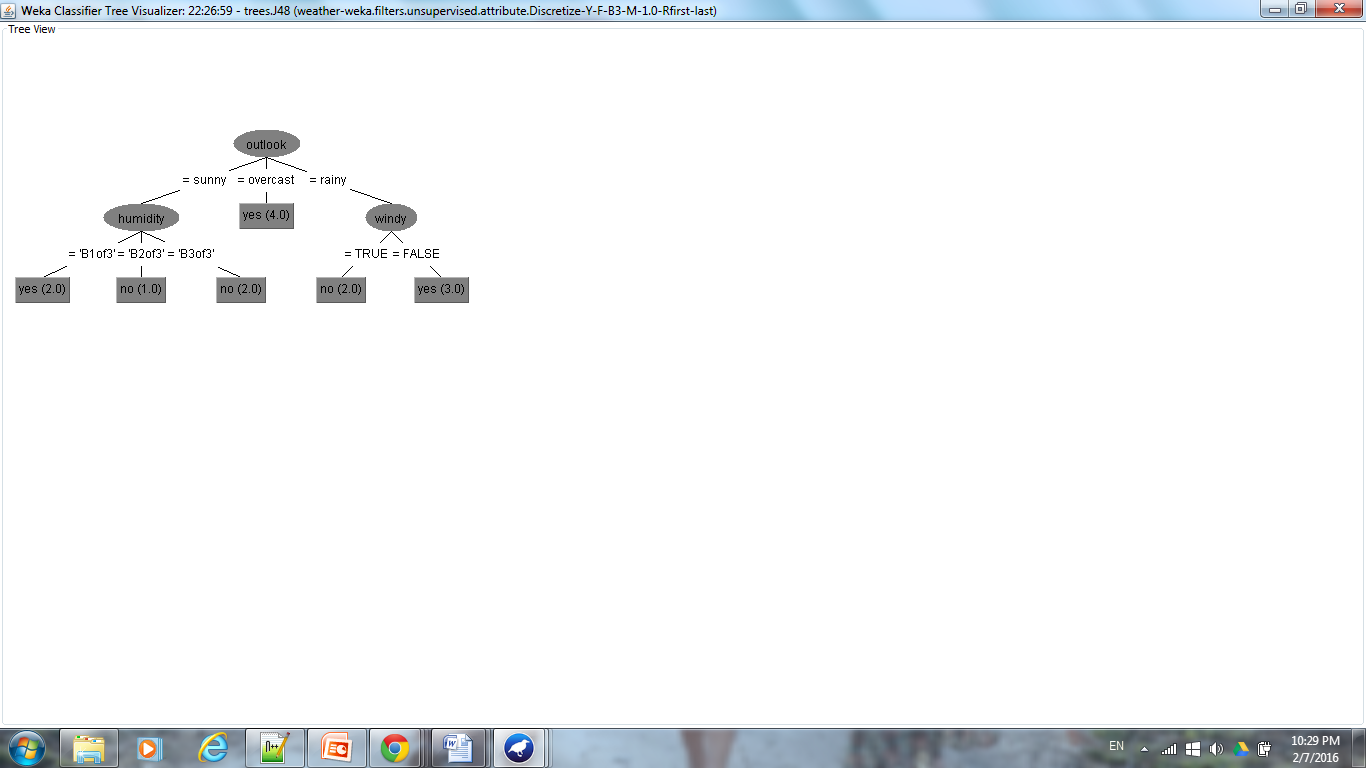
B1. The file Weather. arff was opened in Weka. The data set contains 5 attributes, with supervised learning to be completed in order to determine the outcome of the class attribute “play”, which provides a nominal answer (yes/no) as to whether or not to play base do weather information. In order to be able to work with the real numeric values listed in the dataset, the following filtered were completed. Clicking “unsupervised”, then “discretize”, then “temperature”, and then left clicking onto filter header opens up the popup below. The number of bins was chosen to be 3 in order to allow for at least 1 instance to be placed into each bin (and also to match the division of temperature and huminity values like in weather.nominal.arff), “useBinNumber” was chosen to be True, and “useEqualFrequency” was chosen to be True in order to allow Weka to learn to use the numeric values listed in the data set as nominal values. Selecting the “Apply” button allowed the temperature and humidity numeric attribute values to become placed into 3 discrete bins.



C1. “Visualize All” allowed for the visualization of the attribute distributions over all of the 14 instances. Note: the discretization of the temperature and humidity attributes.



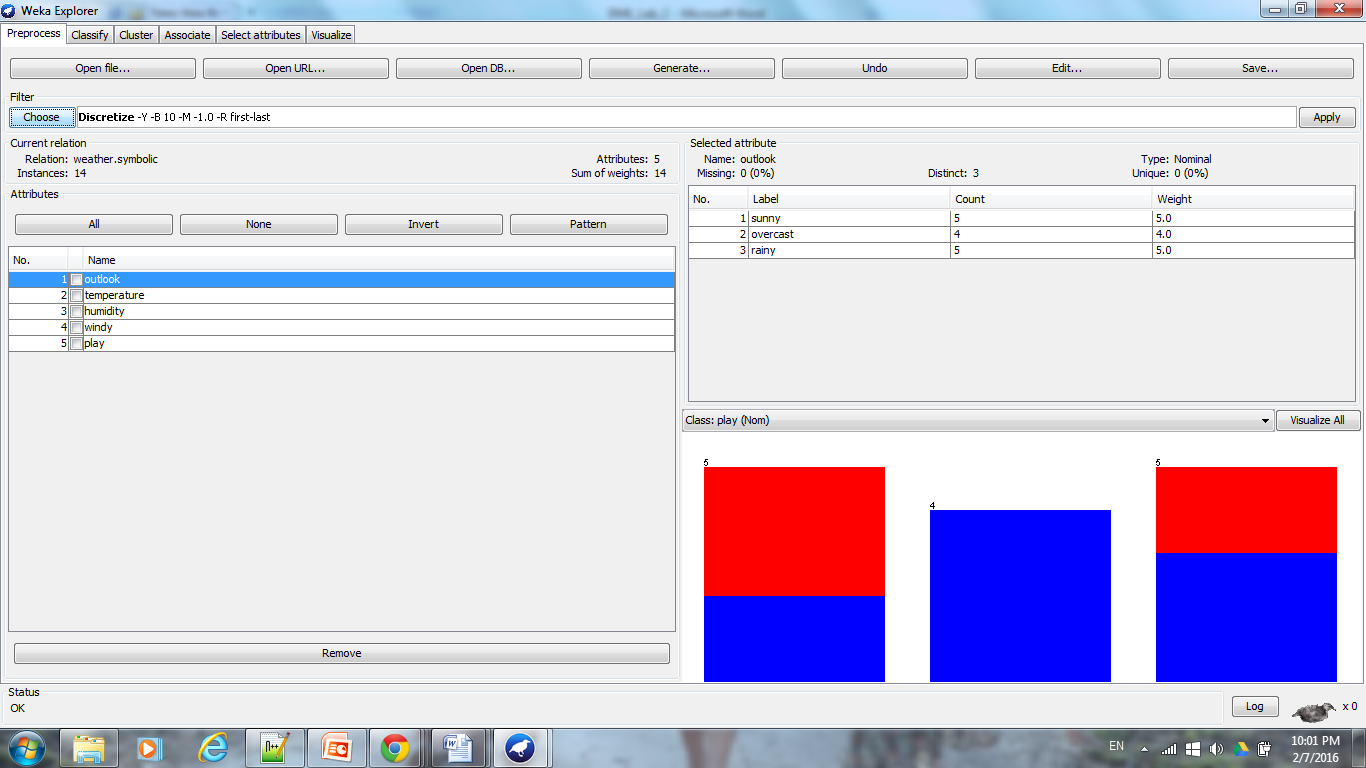
D1. In selecting the “Classify” tab, the J48 decision tree was generated using “Cross validation” at “Fold 10” in order to create the best performing model. It should be noted that B# or bin number was a major indicator as to how to divide the information for humidity.



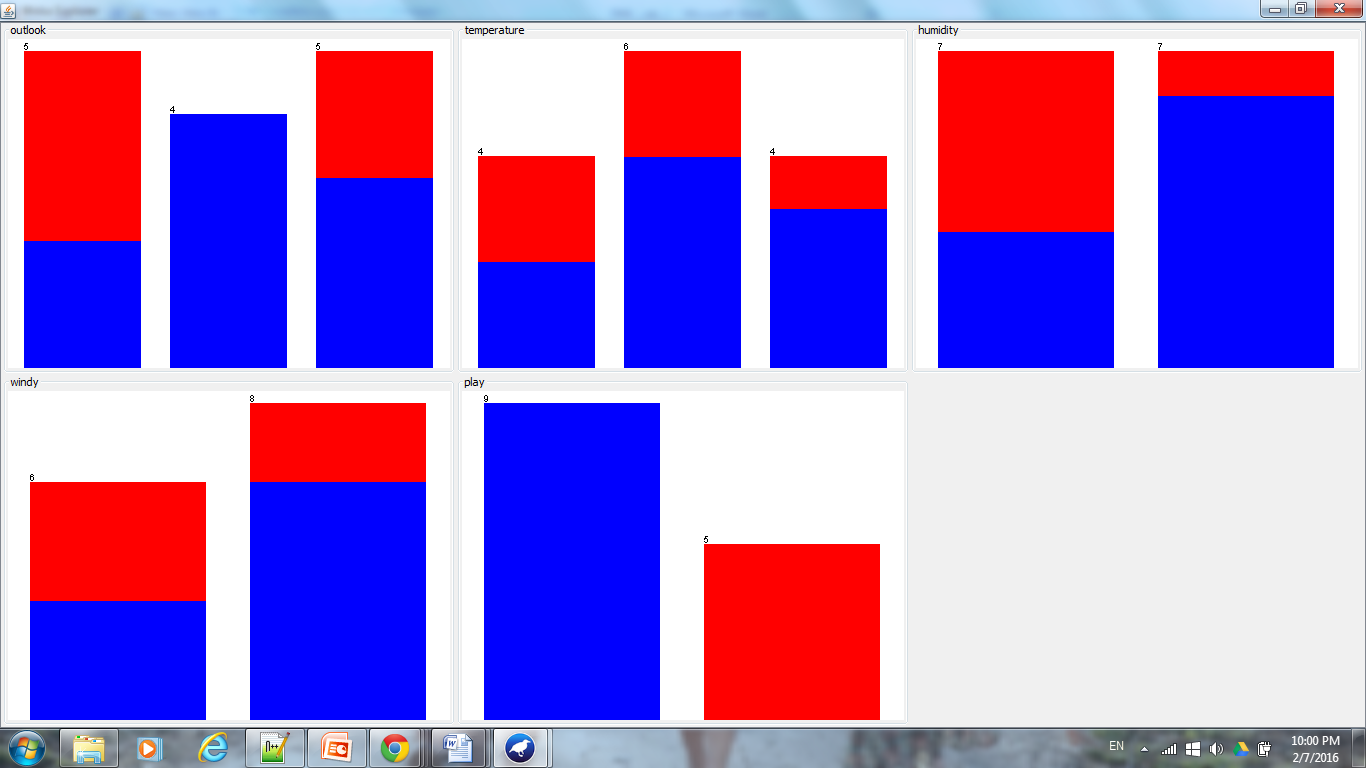
From the generated model and summary information, it can be seen that 6 instances were classified correctly, 8 were classified incorrectly, and the mean absolute error was high 0.4857. This error rate can be potentially decreased by further decreasing by changing the number of discrete bins for humidity and termperature during the “Preprocess” phase. Yet, it should be noted that from repeated testing, it was seen that 3 discrete bins for temperature would provide a potential model for the Weather.arff data set.

Next, the process and analysis above was repeated using the Weather.nominal.arff dataset.

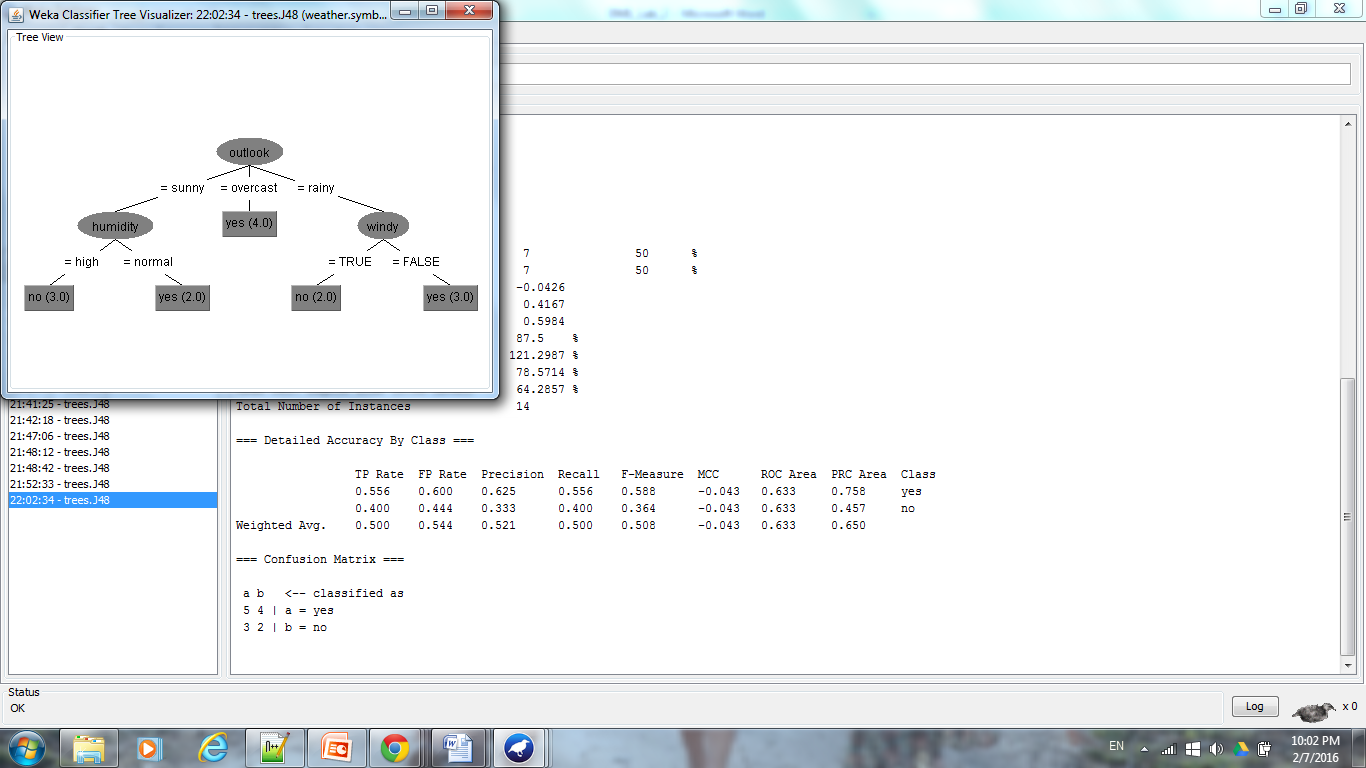
B2. The file Weather.nominal.arff was opened in Weka. The data set contains 5 attributes, with supervised learning to be completed in order to determine the outcome of the class attribute “play”, which provides a nominal answer (yes/no) as to whether or not to play base do weather information. No further filters were necessary to apply, since all of the values are nominal.



C2. Selecting “Visualize All” in order to see the distribution of all the attributes for 14 instances. Note: temperature values are automatically placed into 3 bins based on the nominal value.



D2. In selecting the “Classify” tab, the J48 decision tree was generated using “Cross validation” at “Fold 10” in order to create the best performing model. It should be noted that based on the nominal data set, the numeric dataset can be further discretized for the humidity and temperature values to reach a similar model as shown below.

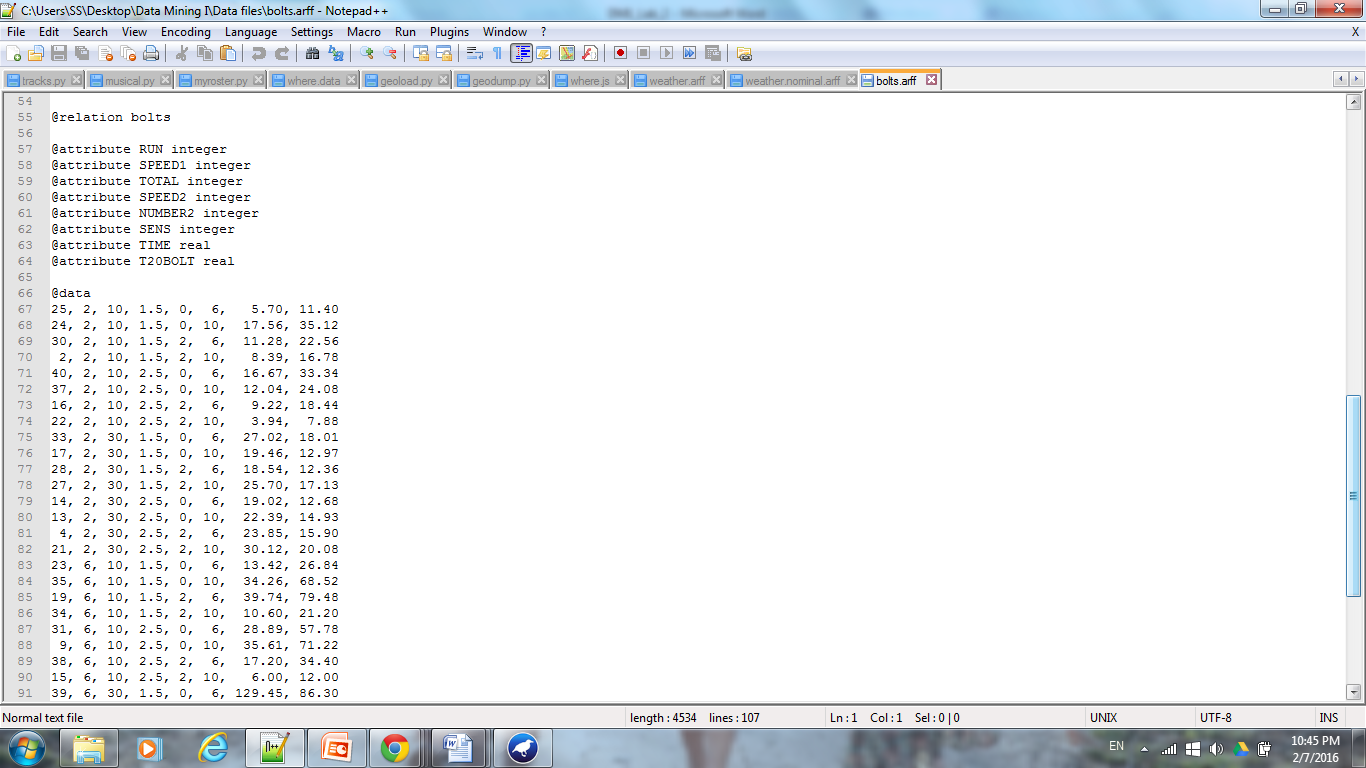


From the generated model and summary information, it can be seen that 7 instances were classified correctly, 7 were classified incorrectly, and the mean absolute error was lower than that of D1 at 0.4167.

What was learned:

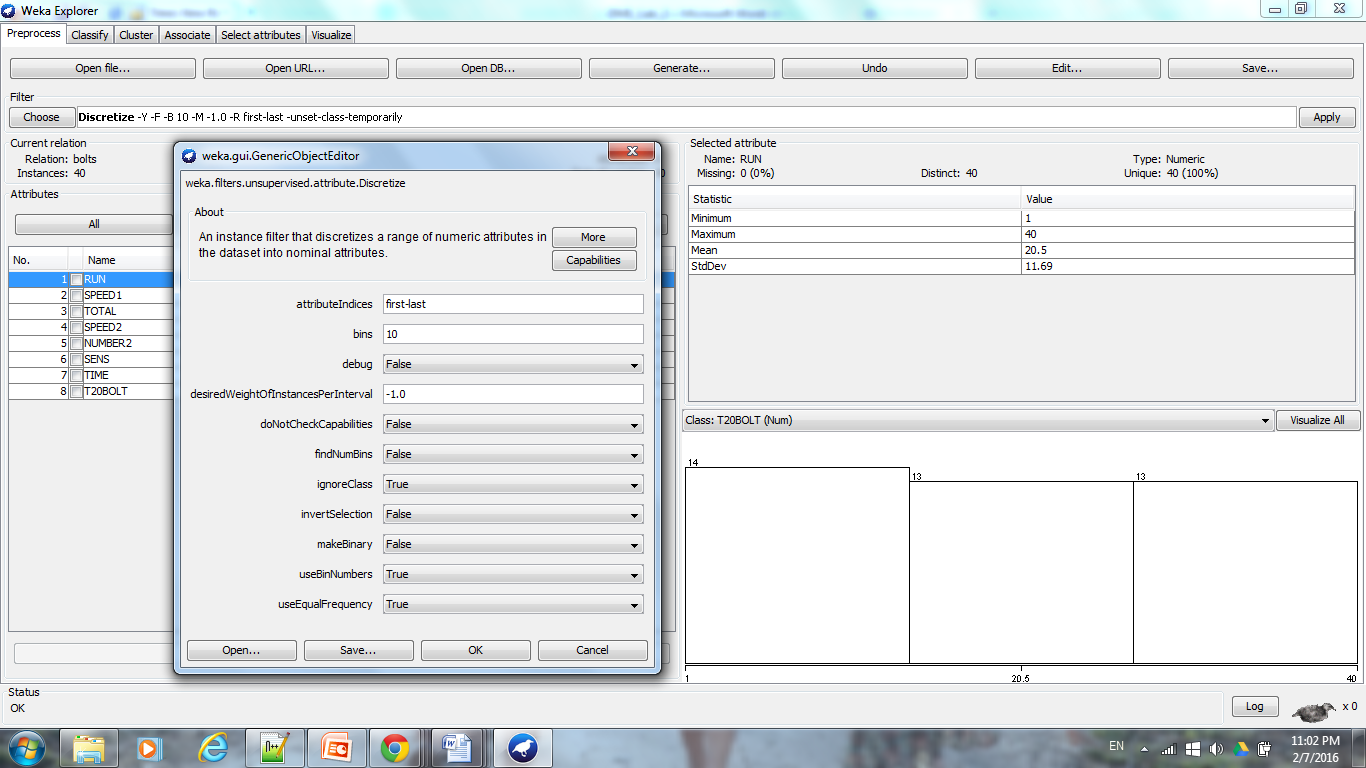
Numeric discretization will allow for Weka to run algorithms with more user control. Bin numbers are essential in determining how the model will compare to that of the pre-conditional nominal data set. Trial and error was used to try to find the best model of the weather.arff data set and analysis the real values of temperature as best as possible through discretization. Temperature was divided into bins for weather.arff depending on user preference, but temperature was divided automatically for the weather.nominal.arff data set.

3.

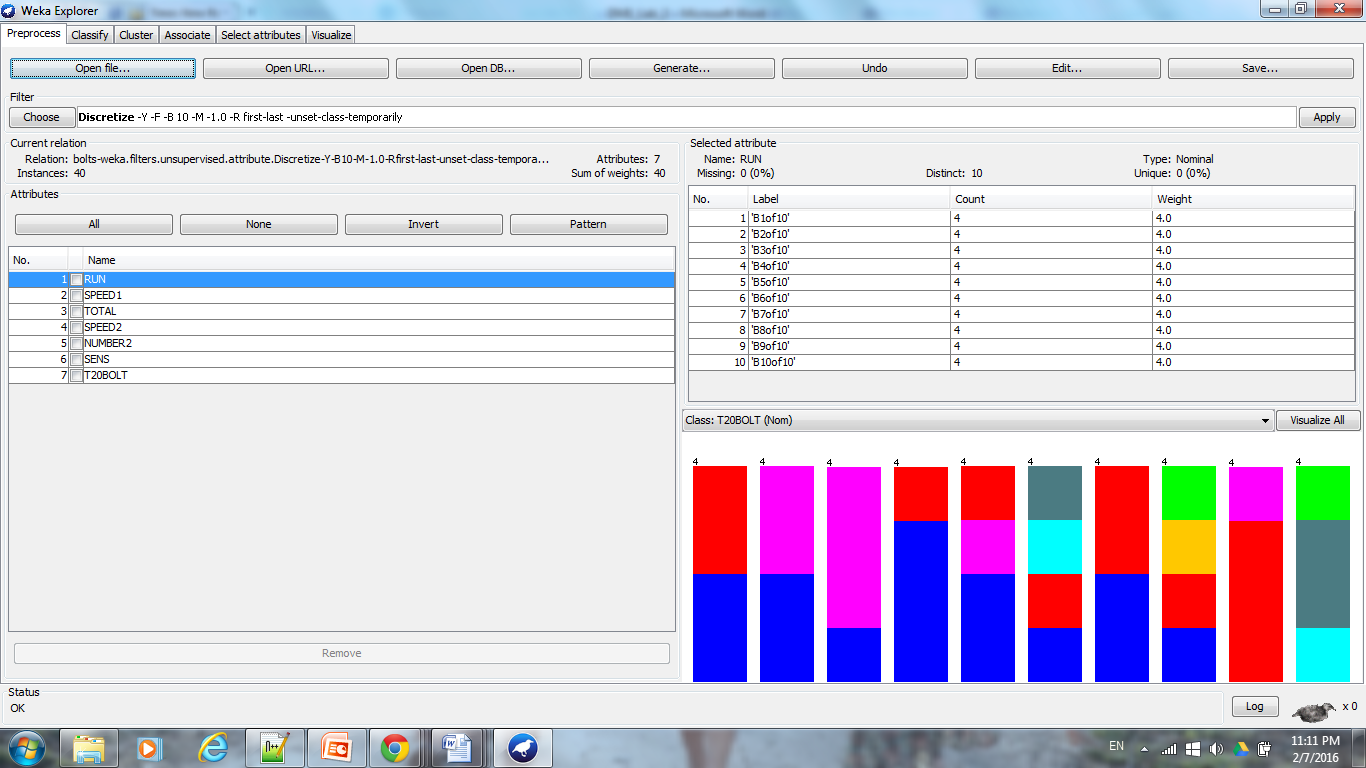
A. Analysis of the bolts.arff data set showed that the attribute values are associated with integer or real values. Therefore, when inputting the data set into Weka, the proper filter parameters must be placed. 

B. The bolts.arff dataset was uploaded into Weka and would be filtered through the “Preprocess” phase. 8 attributes were listed for the data set, all of which are numeric. In order to change the numeric values into nominal values, “unsupervised”, “attribute”, and “discretized” was selected. Next, clicking on the heading produced a popup window (shown below) and the number of bins was selected to be 10, “ignoreClass” was changed to true, and “useBinNumber” was changed to true. Applying the filter onto the data set changed the numeric values to nominal values (as shown below). Finally, “removing” the Time attribute from the data set would create a data set which did not depend on the value of Time to determine the time needed by a machine to produce and count 20 bolts.

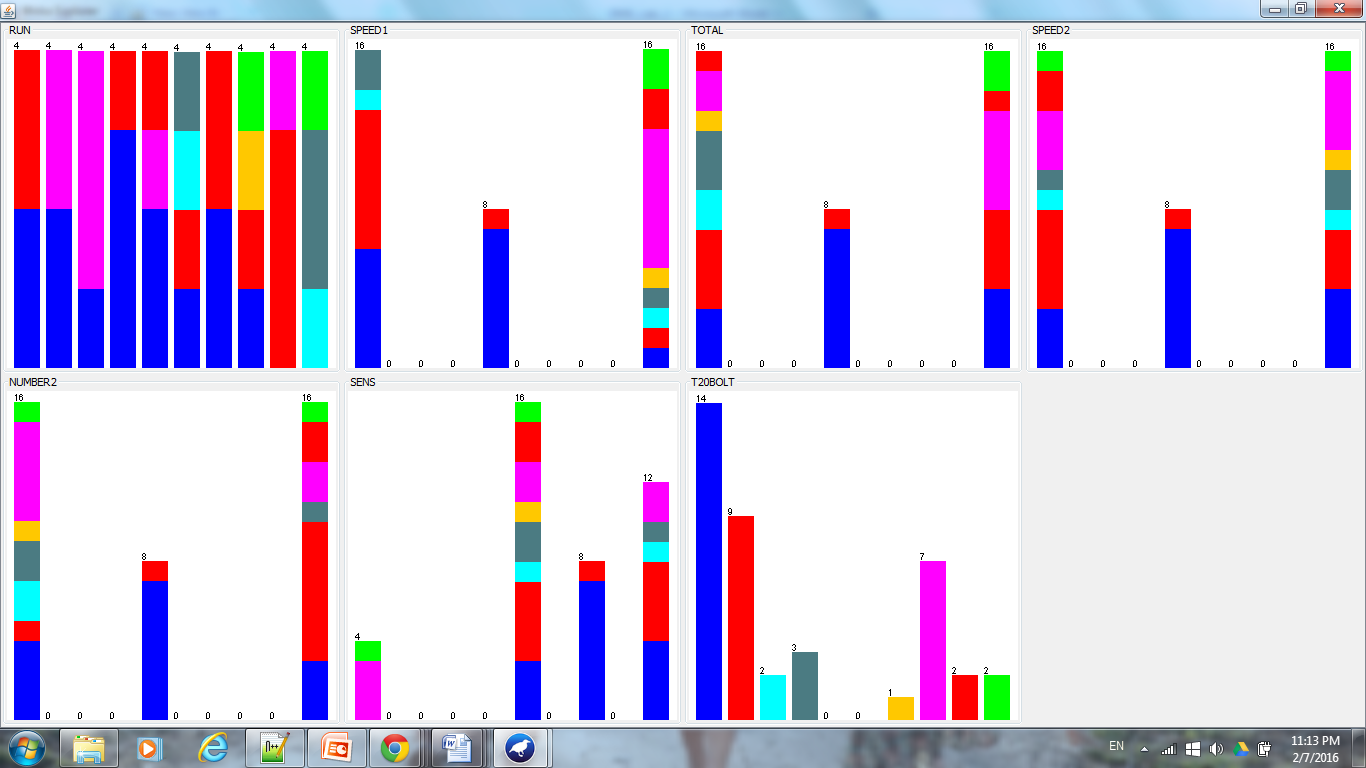
Before applying:



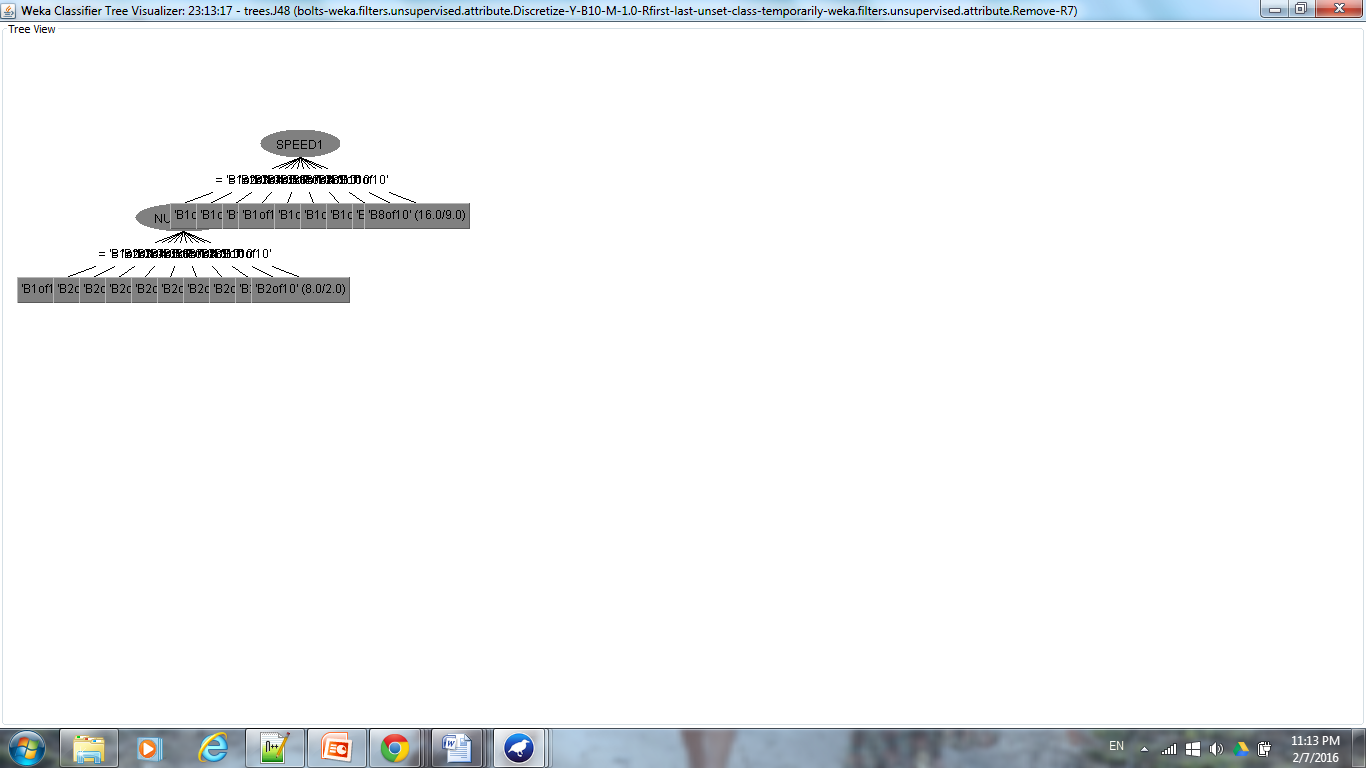
After applying (Note: type of data under “Selected Attribute” changed from “Numeric” to “Nominal”):



C. “Visualize All” showed the attribute distributions for each of the instances. Below, it can be seen that each of the attributes are divided into 10 bins.

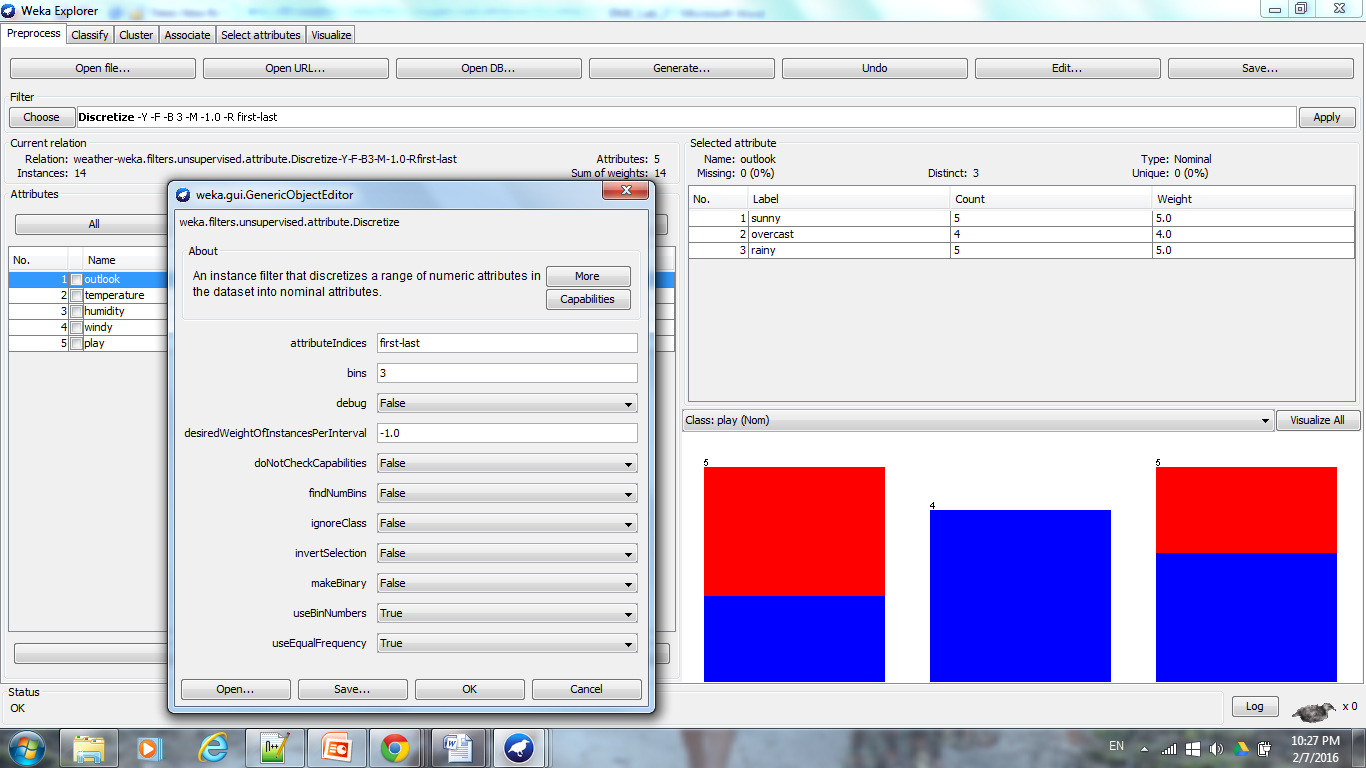


D. Applying the J48 decision tree learning scheme, shows that in the absence of the Time attribute, SPEED1 and Number2 have the largest effect on determining which “Bin#” for T20BOLT an instance will be classified into. Therefore, it can be seen that the speed setting that controls the speed of rotation (SPEED1) of the plate at the bottom of the dish and the number of bolts to be counted at the second speed (NUMBER2) affect T20BOLT the most. According to the decision tree, by shortening the value of SPEED1, the shortest T20BOLT can be achieved.

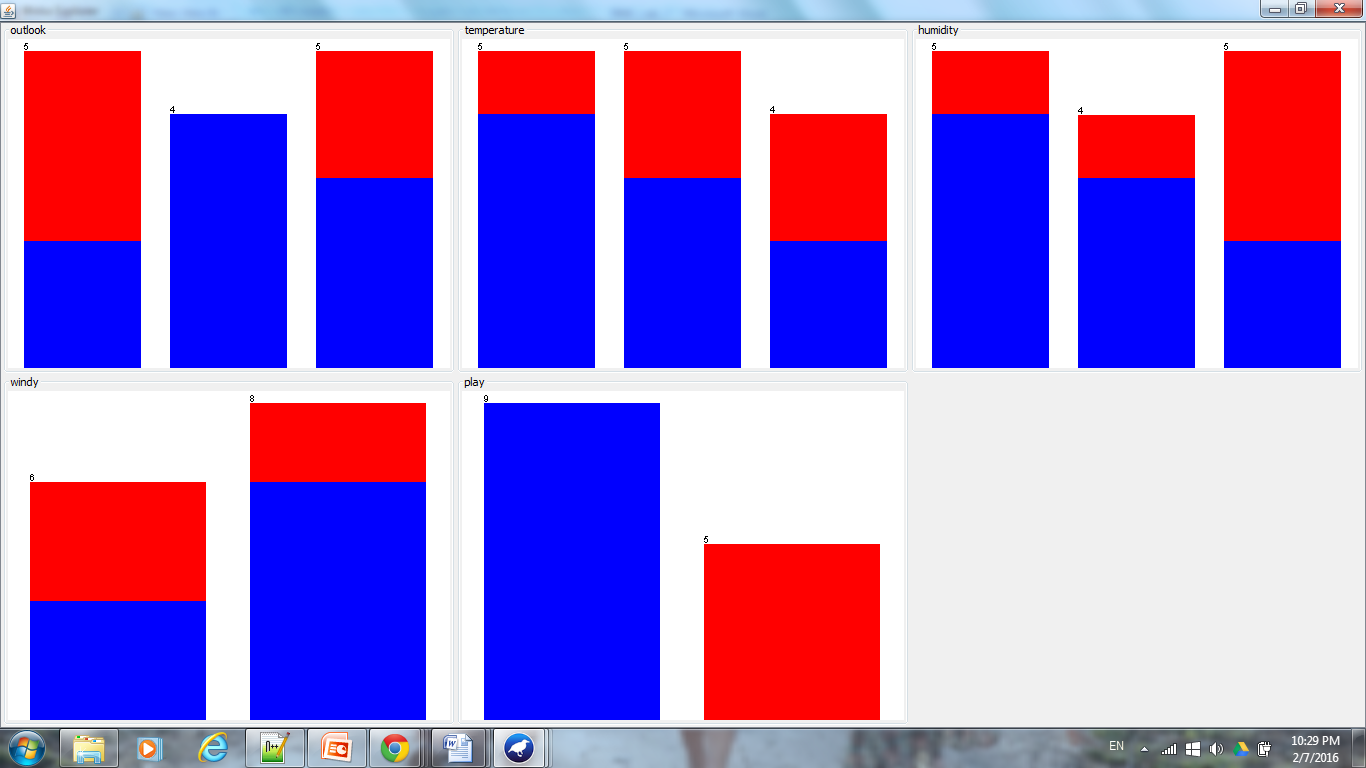


4.

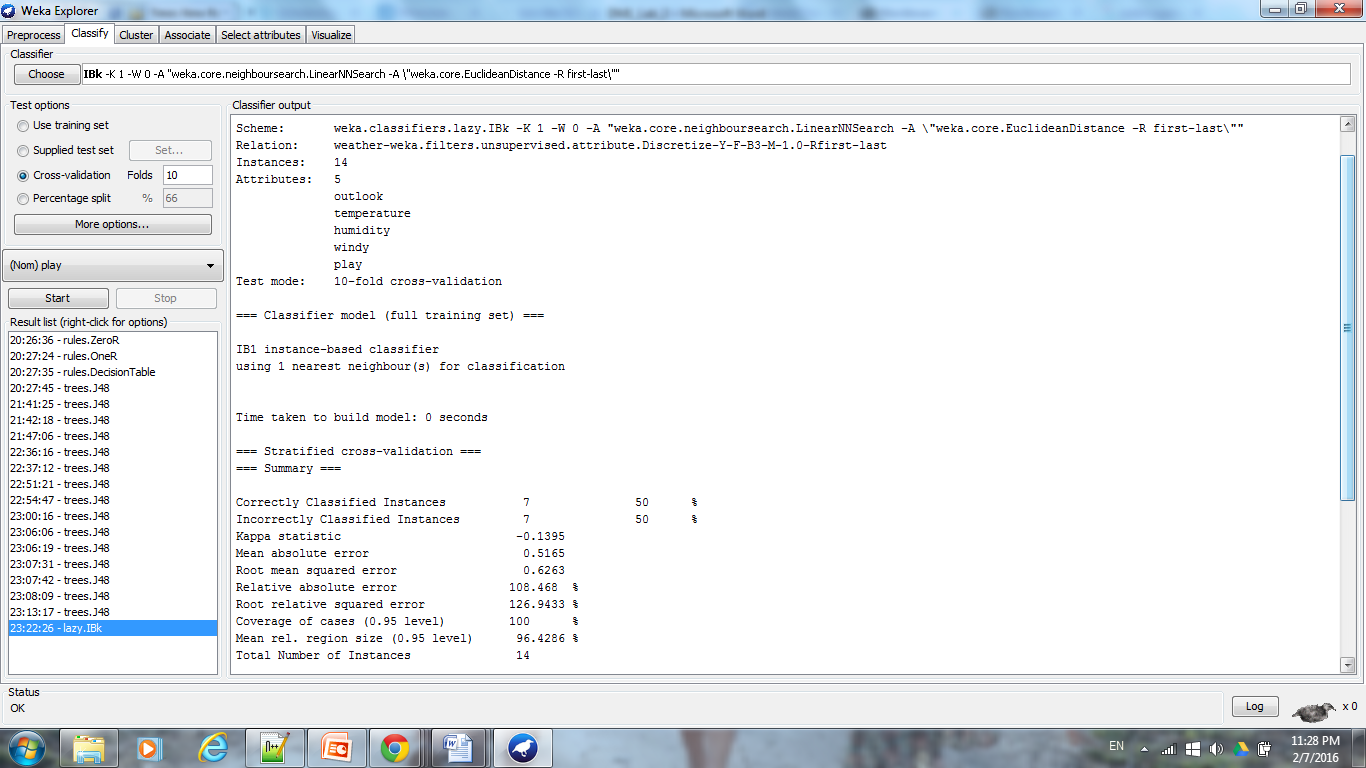
A. The file Weather. arff was opened in Weka. The data set contains 5 attributes, with supervised learning to be completed in order to determine the outcome of the class attribute “play”, which provides a nominal answer (yes/no) as to whether or not to play base do weather information. In order to be able to work with the real numeric values listed in the dataset, the following filtered were completed. Clicking “unsupervised”, then “discretize”, then “temperature”, and then left clicking onto filter header opens up the popup below. The number of bins was chosen to be 3 in order to allow for at least 1 instance to be placed into each bin (and also to match the division of temperature and huminity values like in weather.nominal.arff), “useBinNumber” was chosen to be True, and “useEqualFrequency” was chosen to be True in order to allow Weka to learn to use the numeric values listed in the data set as nominal values. Selecting the “Apply” button allowed the temperature and humidity numeric attribute values to become placed into 3 discrete bins.



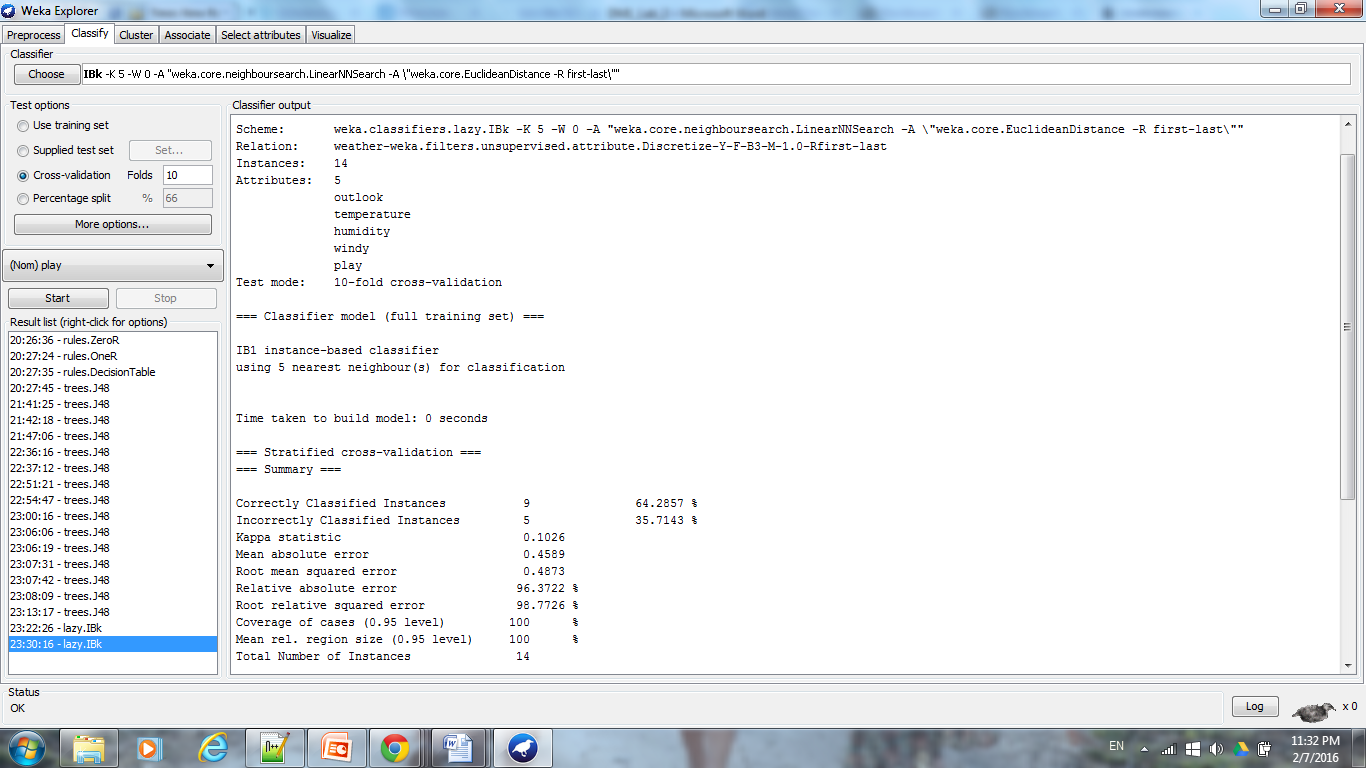
C1. “Visualize All” allowed for the visualization of the attribute distributions over all of the 14 instances. Note: the discretization of the temperature and humidity attributes.



D1. In selecting the “Classify” tab, a K-NN model was produced (classifiers/lazy/IBk). The default K value is set to 1, allowing 7 instances to be correctly classified, 7 instances to be incorrectly classified, and a large mean absolute error of 0.5165.



By increasing the parameter K (the number of neighbors) to 5, results in 9 correctly classified instance, 5 incorrectly classified instances, and a lower mean absolute error of 0.4589.



Therefore, it should be noted that increasing K will decrease variance and increase bias.

But, using different weighting schemes ie. changing “distanceWeighting” to “weight by 1/distance” does not change the output or performance allowing 7 instances to be correctly classified, 7 instances to be incorrectly classified, and a large mean absolute error of 0.5165 if k = 1.

