Data Mining Assignment 3

Ankit Kulkarni

April 17, 2016

I declare that all material in this assessment task is my own work except where there is clear acknowledgement or reference to the work of others and I have complied and agreed to the UIS Academic Integrity Policy at the University website URL: http://www.uis.edu/academicintegrity

Name: Ankita Kulkarni **UID**: 672369586 **Date**: April 16, 2016

1. The ID3 algorithm which is a type of a tree classifier generates a decision tree from the input data set. It is uses an iterative approach for building the decision tree. For the ID3 classifier the tree cannot be visualized. This algorithm uses one of the attribute which has the highest information gain. This attributes forms the root of the decision tree. For each possible value of the root attribute forms the branch of that root. The same pattern is followed again until all the instances are classified. I tested the ID3 algorithm for the weather data set for the test cases Percentage Split and Cross Validation. The test case Percentage Split did not prove to work very good for the weather data and produced a result of 71% when split the data by 70%. When executing the Cross Validation test case for the same algorithm for 10 folds the correctly classified instances were increased to 85%. However the percent of the correctly classified instances decreased when the number of folds were increased or decreased. The best result it could produce was for 10 folds. I could observe that the test case Cross Validation worked better for the weather data set when classified using the ID3 algorithm than percentage split.

   I also experimented the weather data by running the base classifier i.e. ZeroR on it. However the ZeroR classifier could not produce a better result when compared to ID3. It could only correctly classify instances to a maximum of 64% when run for the Cross validation test case and 66% for the test case Percentage Split.

2. As noted earlier, the ID3 classifier cannot be executed on the data set which has missing values. Therefore to run this classifier on the breast-cancer data set the missing values from it should be handled.

   Methodology: There are various method to deal with the missing values. Such as:

   i. Set the missing values to the values which has the maximum frequency

   ii. Set the missing values by comparing the rows. The row which has missing value was compared with all the other rows. The missing value was then set to a value of the row which was maximum similar to the missing value row.

   The heuristic approach that I used is the (iii) one. For implementing this method I used Visual Basic for the application in MS Excel. In the VBA Editor I implemented for loop and a series of If-Then-Else statements to compare each row with the row which has missing value and hence set a value for it. The VBA code was executed twice. In the first run similarity of 5 attributes was checked after which there still existed 2 missing values. Therefore in the second run a similarity of 3 attributes was checked which replaced the remaining 2 missing values as well. After all the missing values were set to some value the excel file was again exported to an arff format and the classifier ID3 was run on it.

   Result: On executing the ID3 classifier on the new arff file the maximum correctly classified instances were 58.042% for 30 folds. Below are the results which were observed for the old breast-cancer dataset and the new breast-cancer dataset:

S. No.	Breast-Cancer Data set		New Breast-Cancer Data set	
	No. of Folds	Correctly Classified Instances (%)	No. of Folds	Correctly Classified Instances (%)
1	10	56.993	10	53.4965
2	20	56.2937	20	56.6434
3	30	57.3427	30	58.042
4	40	58.3413	40	57.6923

It was observed that for the new breast-cancer data set there was a noticeable increase in the percent of classified instances with the increase in the number of folds and then it decreased. Its best performance could be observed when the number of folds were set to 30 and above except for the value 40. However for the original data set the performance of ID3 increased slightly with the increase in number of folds.

3. The ID3 algorithm cannot be executed on the soybean data set directly because the data set does not meet the classifier's capability. The soybean data set contains missing values and the ID3 classifier has a property that it cannot deal with the missing values. Therefore in order to enable the ID3 classifier to run on the soybean data set the "ReplaceMissingValues" filter is needed to be applied. This methodology was implemented just to matchup the soybean data set so that the ID3 algorithm can be run on it.

   Once the ID3 classifier is enabled the next step is to split the data in test and training data. This is done by using the "RemovePercentage filter". The classifier was run for different number of training instances. Below are the observations:

S. No.	Training Instances (%)	Correctly Classified Instances (%)
1	80	88.3212
2	75	81.8713
3	70	86.8291
4	60	82.0513
5	50	83.5777

   The above test were run for the test case Cross validation where the number of folds were set to 10. It was clearly notices that there was no major difference in the result if the number of folds were increased or decreased. The best result was produced by the ID3 classifier when the data set was split for 70% training instances.

   I also tested the data set for the ZeroR classifier however it did not prove to work for the soybean data set. Even after splitting the data set in test and training sets the results were not good for different number of folds. ZeroR could classify the instances correctly maximum up to 35% approximately. The performance of ZeroR did not increases even after dealing with the missing values by implementing the "ReplaceMissingValues" filter.

4. Any data set can be classified by dividing the data set into a small train data and a large test data, where the test data is supplied as a test option for classifying. Here the soybean data set was divided into a small train data and a large test data. The percentage used for splitting was 70-30%, where 70% was test data and 30% was train data. Also for

this experiment the "Output Prediction" was enabled. On running the ID3 classifier for this split the correctly classified instances were 38.2845%. To increase the percentage of the correctly classified instances the incorrectly classified instances were removed from the test data and were added to the train data and the increase in the performance could be observed. This procedure was carried out 5 times.

In the first step 6% of the test data which included only the incorrectly classified instances was removed from the test data set and added to the train data set. In the next step 7% of the remaining test data was removed and added to the training data set. Similarly 8% was removed followed by 12% and finally 5%. However after implementing this iterative method to add the incorrectly classified instances to the train data, it was clearly observed that this method was able to train the data more quickly than by directly training using the entire training data set. In the method where the data set is trained directly from the entire training data set the result will be good because the entire data set is provided as training data. Also using the cross validation test case will produce a good result because there the classifier remembers the data and hence its performance is good.

5. With the above iterative method there was a significant increase in the percentage of the correctly classified instances. Below are the results that were observed when this iterative method was used to train the data:

S. No.	Percent of incorrectly classified instances moved from test data and train data	No. of incorrectly classified instances removed	No. of instances in test data set	Correctly classified instances (%)
1	6	29	499	45.657
2	7	31	418	52.6316
3	8	33	385	60.7792
4	12	46	339	80.531
5	5	16	323	85.4037

6. The User Classifier is a type of tree classifier which is interactive and allows user to build their own decision tree. To classify data using the User Classifier the data set first needs to be loaded in Weka and then the User Classifier is to be selected. For implementing the User classifier on a data set, the test case "Percentage split" is used

with 66%. Here in this experiment the soybean data set is split into test and training data. The soybean data without missing values is loaded in Weka and the test case “Percentage Split” is set to 66%. After setting up the above properties when the classifier is run, a Tree Visualizer and Data Visualizer is opened. In the data visualizer the data points on the graph are selected using a rectangle and are removed by submitting it to the tree. The tree formed can be seen in the Tree Visualizer. Below is the result and the tree that was produced at my end for the User Classifier:

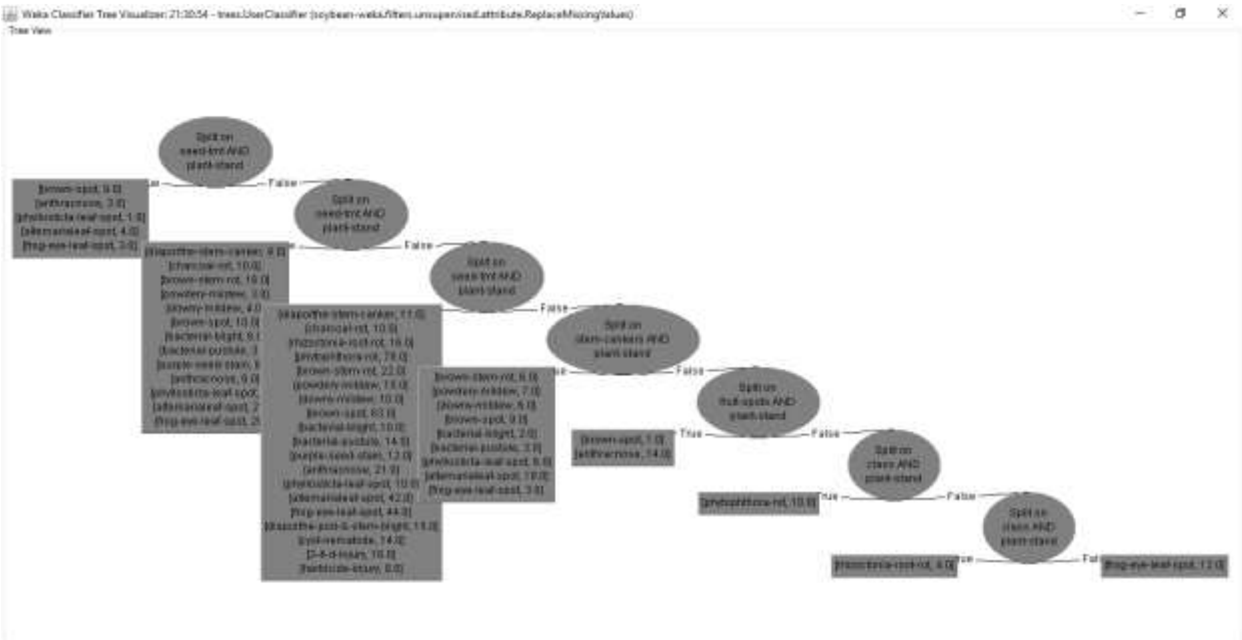
Classifier output

```
Split on seed-tmt AND plant-stand (In Set): N1 brown-spot(20.0/11.0)
Split on seed-tmt AND plant-stand (Not in Set)
|   Split on seed-tmt AND plant-stand (In Set): N3 frog-eye-leaf-spot(136.0/107.0)
|   Split on seed-tmt AND plant-stand (Not in Set)
|   |   Split on seed-tmt AND plant-stand (In Set): N5 phytophthora-rot(426.0/348.0)
|   |   Split on seed-tmt AND plant-stand (Not in Set)
|   |   |   Split on stem-cankers AND plant-stand (In Set): N7 alternarialeaf-spot(60.0/42.0)
|   |   |   Split on stem-cankers AND plant-stand (Not in Set)
|   |   |   |   Split on fruit-spots AND plant-stand (In Set): N9 anthracnose(15.0/1.0)
|   |   |   |   Split on fruit-spots AND plant-stand (Not in Set)
|   |   |   |   |   Split on class AND plant-stand (In Set): N11 phytophthora-rot(10.0)
|   |   |   |   |   Split on class AND plant-stand (Not in Set)
|   |   |   |   |   |   Split on class AND plant-stand (In Set): N13 rhizoctonia-root-rot(4.0)
|   |   |   |   |   |   Split on class AND plant-stand (Not in Set): N14 frog-eye-leaf-spot(12.0)

Time taken to build model: 61.66 seconds

=== Evaluation on test split ===
=== Summary ===

Correctly Classified Instances          93               40.0862 %
Incorrectly Classified Instances       139               59.9138 %
Kappa statistic                          0.3136
Mean absolute error                      0.0747
Root mean squared error                  0.1932
Relative absolute error                 77.6705 %
Root relative squared error             88.2461 %
Total Number of Instances              232
```



The classification that was performed through this method at my end could produce a maximum 40.0862% percent of correctly classified instances.