Z Number: 23226341

Name: Carlos Gross-Martinez

Professor: Dr. Taghi M. Khoshgoftaar

Course: CAP6778 – Advanced Data Mining & Machine Learning

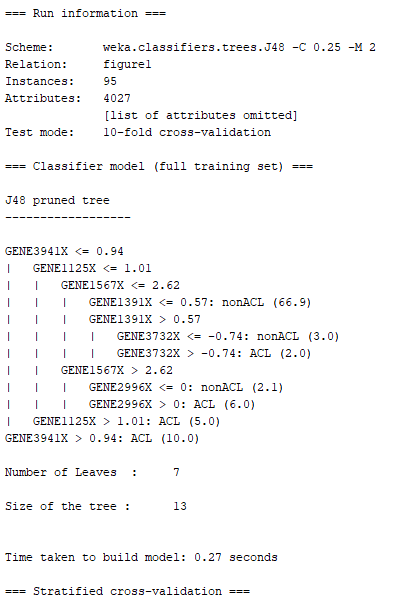
Assignment: Modeling Assignment: Classification Using Decision Trees

Dataset Analysis

The dataset used in this assignment consisted of 95 instances/samples and 4027 attributes. From the 4027 attributes 4026 were input attributes to the model while 1 attribute was the class label. Furthermore, out of the 95 samples provided in the dataset, 23 were from the minority class labeled “ACL” while 72 were from the majority class labeled “nonACL”. With this information, it can be noted that percentage wise, the minority class represents of the full dataset, while the majority class represents of the full dataset. With this data distribution in mind is important to note that the data set contains contain high dimensionality due to the high number of attributes serving as an input as well as class imbalance, as the data contains a great number of samples for the secondary class “nonACL” while containing a smaller number of samples for the primary class “ACL”.

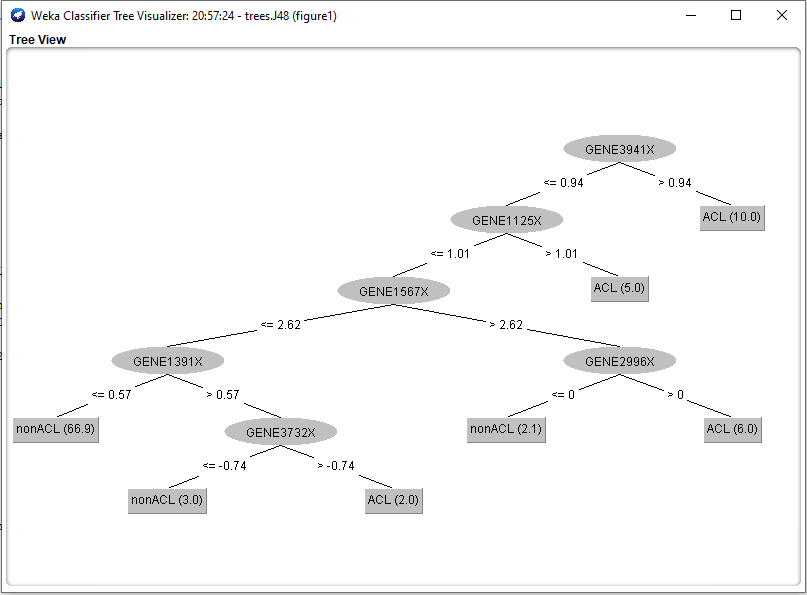
Part 1: Initial Tree

In this segment of the assignment a classification model is built using the J48(C4.5) decision tree algorithm with 10-fold cross validation with the unpruned option set to false. A picture of the trained model can be observed below:



Picture 1: J48 Pruned Decision Tree Results

The implementation of the J48 ML algorithm created a decision tree with 6 nodes and 7 leaves. Picture of the decision tree created by the algorithm can be seen below:



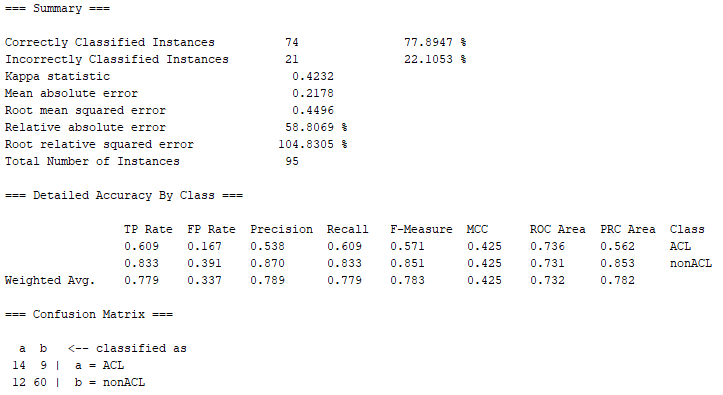
Picture 2: J48 Pruned Decision Tree

The confusion matrix for the J48 decision tree can be observed below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted | | |
| Actual |  | ACL | nonACL |
| ACL | 14  (TN) | 9  Type II (FN) |
| nonACL | 12  Type I (FP) | 60  (TP) |

Table 1: Confusion Matrix J48 Pruned Decision Tree

To corroborate the above information, a picture displaying the results from the model after it has been trained can be observed below:



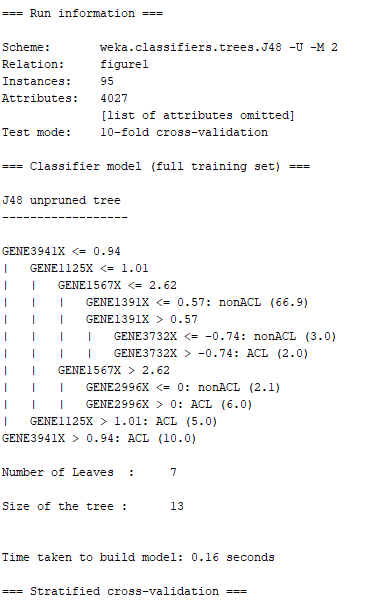
Picture 3: J48 Pruned Results After Training

The next step, consist in calculating the misclassification error rates for both types of misclassifications of the confusion matrix in the model. These calculations can be observed below:

The final step of this segment consists of including on the AUC for the ROC curve. Based on the results provided from the Weka tool, the ROC area for the ACL class is 0.736 while the ROC area for the nonACL class is 0.731.

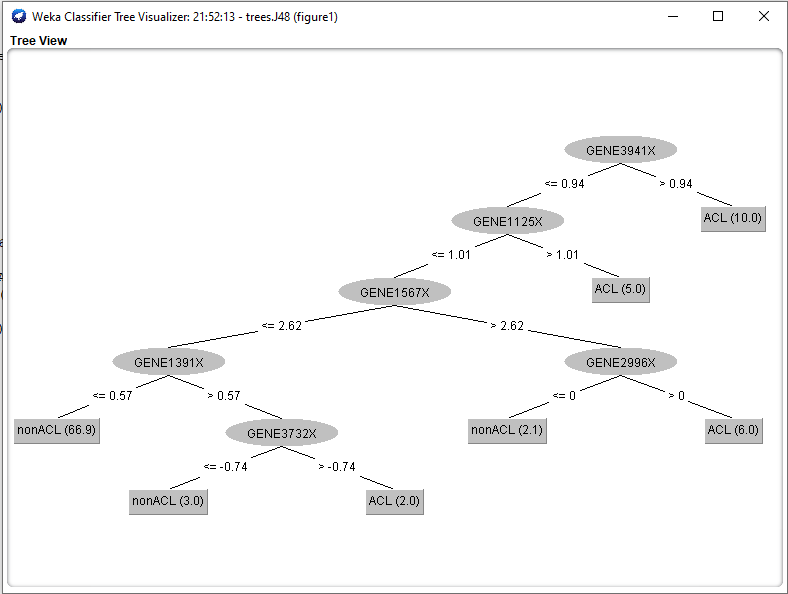
Part 2: Unpruned Tree

In this segment of the assignment a classification model is built using the J48(C4.5) decision tree algorithm with 10-fold cross validation with the unpruned option set to true. A picture of the trained model can be observed below:



Picture 4: J48 Unpruned Decision Tree Results

The implementation of the J48 ML algorithm created a decision tree with 6 nodes and 7 leaves. Picture of the decision tree created by the algorithm can be seen below:



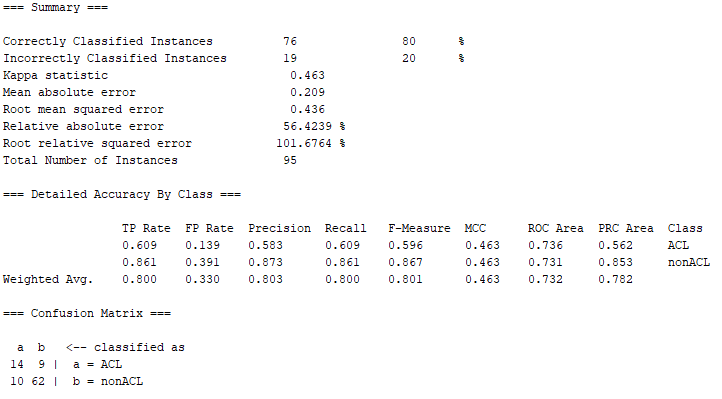
Picture 5: J48 Unpruned Decision Tree

The confusion matrix for the J48 decision tree can be observed below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted | | |
| Actual |  | ACL | nonACL |
| ACL | 14  (TN) | 9  Type II (FN) |
| nonACL | 10  Type I (FP) | 62  (TP) |

Table 2: Confusion Matrix J48 Unpruned Decision Tree

To corroborate the above information, a picture displaying the results from the model after it has been trained can be observed below:



Picture 6: J48 Unpruned Results After Training

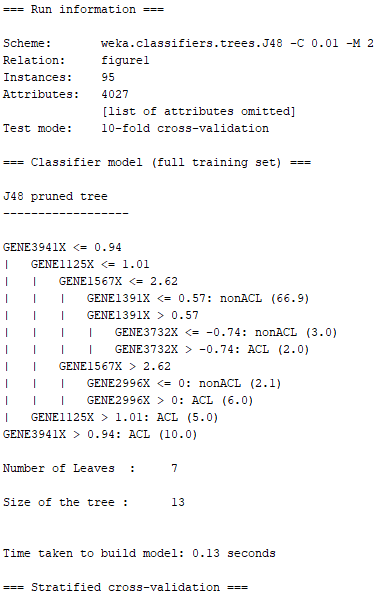
The next step, consist in calculating the misclassification error rates for both types of misclassifications of the confusion matrix in the model. These calculations can be observed below:

The next step of this segment consists of including on the AUC for the ROC curve. Based on the results provided from the Weka tool, the ROC area for the ACL class is 0.736 while the ROC area for the nonACL class is 0.731.

Based on these results from the prune and unpruned model, it can be noted that they both have the same number of nodes and leaves. There was no difference in the overall design from both decision trees. Furthermore, the ROC area for both models remained the same for both classes as well. Moreover, the prune tree had a higher FRP at 0.46 while the unpruned tree had an FRP of 0.42. Additionally, the FNR for both decision trees, prune and unpruned remained constant between both models. Finally, it is important to note that the classification for the primary class (ACL) remained constant between the two models. Nevertheless, the unpruned model performed a little better than its counterpart when classifying the secondary class by properly classifying two more samples in comparison to the prune decision tree. Therefore, the second model had a better performance in classifying the dataset by correctly classifying 76 samples and misclassifying 19 samples. While the prune model properly classified 74 samples and misclassified 21 samples from the dataset.

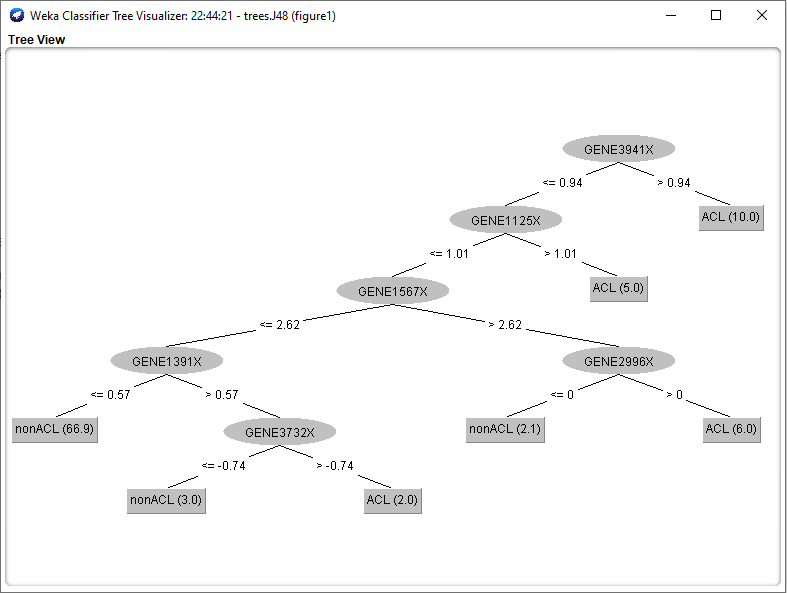
Part 3: Confidence Factor

In this segment of the assignment a classification model is built using the J48(C4.5) decision tree algorithm with 10-fold cross validation with the unpruned option set to false and a confidence factor (C) of 0.01. A picture of the trained model can be observed below:



Picture 7: J48 pruned (C=0.01) Decision Tree Results

The implementation of the J48 ML algorithm created a decision tree with 6 nodes and 7 leaves. Picture of the decision tree created by the algorithm can be seen below:



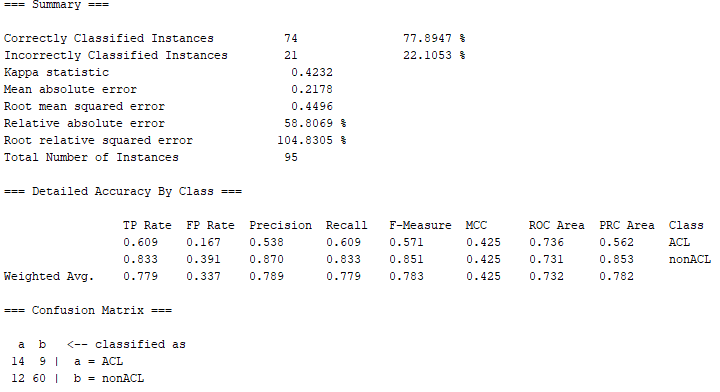
Picture 8: J48 Pruned (C=0.01) Decision Tree

The confusion matrix for the J48 decision tree can be observed below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted | | |
| Actual |  | ACL | nonACL |
| ACL | 14  (TN) | 9  Type II (FN) |
| nonACL | 12  Type I (FP) | 60  (TP) |

Table 3: Confusion Matrix J48 Pruned (C=0.01) Decision Tree

To corroborate the above information, a picture displaying the results from the model after it has been trained can be observed below:



Picture 9: J48 Pruned (C=0.01) Results After Training

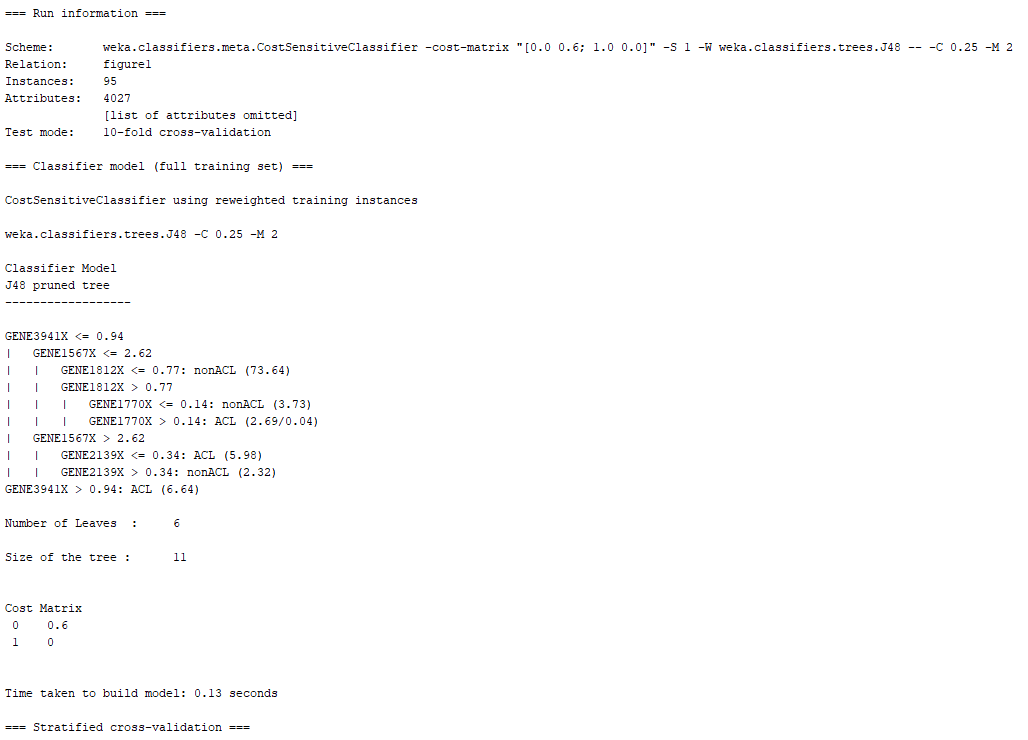
The next step, consist in calculating the misclassification error rates for both types of misclassifications of the confusion matrix in the model. These calculations can be observed below:

The next step of this segment consists of including the AUC metrics for the ROC curve. Based on the results provided from the Weka tool, the ROC area for the ACL class is 0.736 while the ROC area for the nonACL class is 0.731.

Based on these results it can be noted that there is no difference of design between the J48 pruned decision tree with a confidence factor of 0.01 and the decision tree from part 1. The results obtained through Weka demonstrates that both trees had the same number of nodes and leaves. Additionally, all attributes on each node to construct the trees were the same for part 1 and 3. This is also true for the metrics of the leaf nodes as well as the numerical values used to transverse the node to arrive at the leaf nodes. Due to this equality, both trees obtained the same confusion matrix, FPR, FNR, and ROC are values. With this information in mind, no nodes nor leaves from the current tree were severe in comparison to the decision tree constructed in part 1. I believe that the main reason why this occurred is because the confidence factor was dramatically reduced to favor the majority class, is because of the high dimensionality as well as the class imbalance of the dataset. From all parts of the assignment so far, it can be noted that the minority class was classified exactly the same in all three scenarios. With this in mind, it is important to note that when calculating the classification error from each class, the minority class always had a higher misclassification rate when compared to the majority class. In the minority class, out of the 23 samples, only 14 samples were properly classified while 9 samples were misclassified. This means that of the samples in the minority class were properly classified while of the samples were mistakenly classify. On the other hand, the majority class properly classified 60 samples while only 12 samples were misclassified. This means that of the samples in the majority class were properly classified while of the samples were mistakenly classified. Based on this information, it can be concluded that majority class had a better individual classification rate due to the fact that the model could better learn the features of the attributes from the majority class since it had more samples to learn from. Furthermore, the high dimensionality of the dataset also played a major factor since it allowed to pick more attributes which represented the characteristics of the majority class.

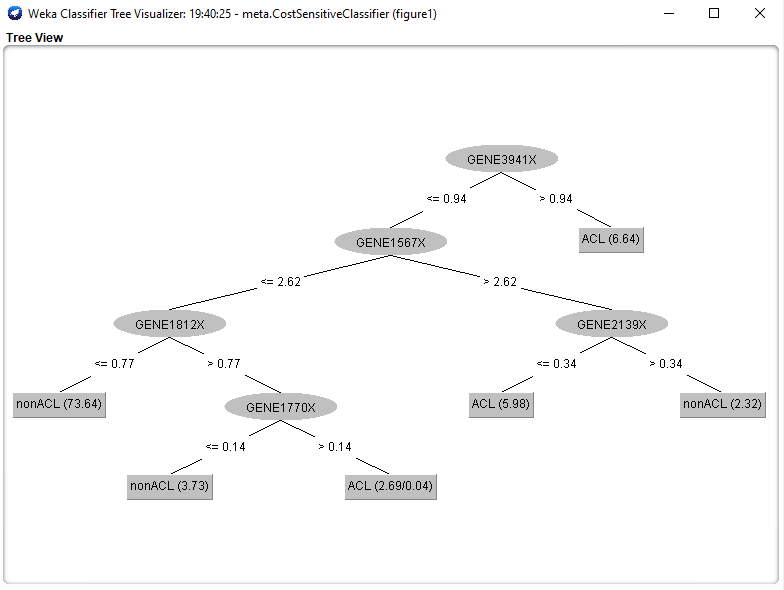
Part 4: Cost Sensitivity

In this segment of the assignment a classification model is built using a cost sensitive classifier with a J48(C4.5) decision tree algorithm with 10-fold cross validation with the unpruned option set to false. Various models were built in order to determine the best optimal cost ratio between the majority and minority classes. A picture of the trained model with the optimal cost ratio can be observed below:



Picture 10: Cost Sensitive J48 pruned Decision Tree Results

The implementation of the cost sensitive classifier with the J48 ML algorithm created a decision tree with 5 nodes and 6 leaves. Picture of the decision tree created by the algorithm can be seen below:



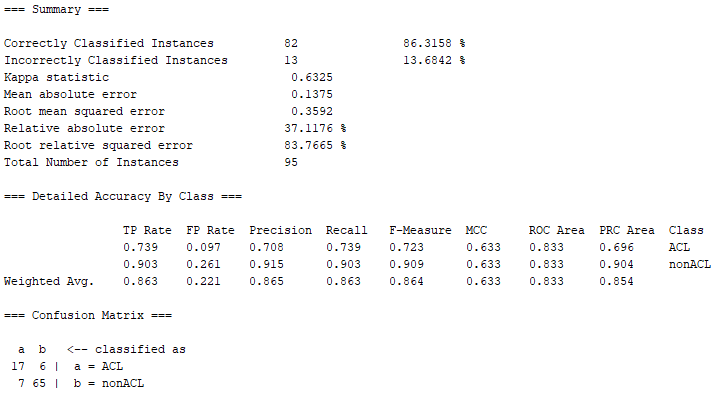
Picture 11: Cost Sensitive J48 Pruned Decision Tree

The confusion matrix for the cost sensitive J48 decision tree can be observed below:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Predicted | | |
| Actual |  | ACL | nonACL |
| ACL | 17  (TN) | 6  Type II (FN) |
| nonACL | 7  Type I (FP) | 65  (TP) |

Table 4: Confusion Matrix Cost Sensitive J48 Pruned Decision Tree

To corroborate the above information, a picture displaying the results from the model after it has been trained can be observed below:



Picture 12: Cost Sensitive J48 Pruned Results After Training

The next step, consist in calculating the misclassification error rates for both types of misclassifications of the confusion matrix in the model. These calculations can be observed below:

The next step of this segment consists of including the AUC metrics for the ROC curve. Based on the results provided from the Weka tool, the ROC area for the ACL class is 0.833 while the ROC area for the nonACL class is 0.833.

Based on the different models that were built under this section of the assignment, the optimal cost sensitive value which resulted in the lowest misclassification for the primary/minority class was 0.6 for the type II error. As the cost sensitive value increased or decreased from the 0.6 threshold, the model increased the misclassification of the samples from the primary class. It is important to note that as the cost sensitive value decrease, the number of samples properly classified from the majority class increased as well. When the cost sensitive value in the cost matrix for type II error decrease to 0, all samples from the dataset were classified as nonACL. As the cost sensitive value increased, the number of misclassifications from the minority class started to decrease until reaching the 0.6 threshold value. Finally, when after the cost sensitive value surpass the threshold, the number of misclassifications of the minority class started to increase as well. It is important to note when the cost sensitive value was set to 0, all samples in the dataset were classified as nonACL. Nevertheless, it did not matter how much the cost sensitive value was increase after the threshold, it was not possible to get all samples in the dataset to be labeled as ACL. Various models with different cost sensitive values provided different ratios of misclassification for both classes. Nonetheless, the optimal cost ratio which resulted in the lowest number of misclassifications for both classes was 0.6.

Based on the results from all the sections in the assignment, it can be concluded that having the ability to tune a model with a cost sensitive factor can greatly improve the classification of a class for which otherwise will not be able to occur through regular machine learning algorithms.