CS584 HOMEWORK 3
NAME : Preeti Bhattacharya (G01302375)
USERNAME IN MINER : Preeti, **ACCURACY SCORE**: 0.95, **RANK**: 8 (At time of Upload)

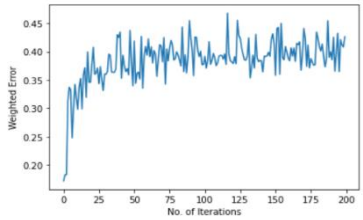
Introduction:
"Boosting" is a way for boosting the performance of any learning algorithm in general. In theory, boosting can be used to significantly reduce the error of any “weak” learning algorithm that consistently generates classifiers which need only be a little bit better than random guessing. Despite the genuine practical value of boosting can only be determined by considering the prospective benefits of boosting as promised by theoretical outcomes and evaluated by putting the method to the test on real-world machine learning issues. In this assignment we present here AdaBoost, a boosting algorithm and a Gini-index (as weak learner) based method for learning decision stumps.

Approach:
I spent the initial days trying to learn about different libraries, what they do and how they are implemented, as I have very basic knowledge about data mining.

1) **Data framing & splitting**:-I dropped all the null values and separated “label” which contained 5 and 3 values in y and the rest in x_data. Then I replaced 5 with 1 and 3 with -1 for easy understanding. I divided the dataset into two sections, with 80 and 20 percent for training and testing.
2) **Training Model**:- I rolled out our own AdaBoost Classifier. Then I have calculated Gini index for every feature by calculating Gini index for each column from where I can choose mini-Gini impurity column. Then I’m iterating through the no of iterations and used decisionStumpClassifier which has min Gini impurity.

$$Gini = 1 - \sum_j p_j^2$$

3) After that I’m fitting the model by finding weak learner. So, I can make a final prediction by taking a “weighted majority vote”, calculated as the sign (±) of the linear combination of each stump’s prediction and its corresponding stump weight.
4) Then I evaluated the performance of our train data based on different iterations, where for 200 iterations I’m getting train error as 1.6% and plotted a graph of weighted error and 200 iterations.
5) After training my model I’m taking test data and removing null and replacing 1 and -1 with 5 and 3 respectively which we converted for our ease.
6) **Output**- Once converted we are writing this data into a txt file and uploading it to miner to get the accuracy of the test data.



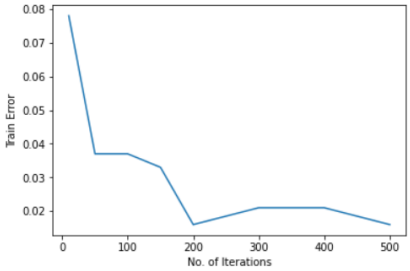
Train error: 1.6%

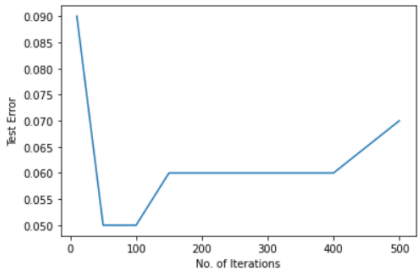
7) **Observation-** After getting the train and test error data which is shown in table below. I have plotted the graph accordingly.

Iterations	Training Error	Test Accuracy	Test Error
10	0.078	0.91	0.09
50	0.037	0.95	0.05
100	0.037	0.95	0.05
150	0.033	0.94	0.06
200	0.016	0.94	0.06
300	0.021	0.94	0.06
400	0.021	0.94	0.06
500	0.016	0.93	0.07

* For the graph 1 and 2 while applying AdaBoost on training data decision stump I started with 10 iterations and found that error rate was 0.078 whereas the accuracy on test data was 0.91. therefore, I got the test error as 0.09 which is greater than training error.
* When I increased the iterations to 50 then I found out that error rate was 0.037 whereas the accuracy on test data was 0.95 in miner therefore, I got the test error as 0.05 from which we can say that the test and train error reduced a lot when compared with iteration 10.
* When I did 100 iteration training error was 0.037 and had accuracy of 0.95 which yielded test error to be 0.05 which was same as earlier and got the lowest error. This range is the good level of fit as per the observation from the table and graph.

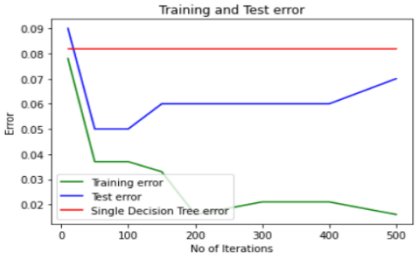
*And when I further boosted the iteration to 150 the training error rate was 0.033 whereas the accuracy on test data was 0.94 and, I got the test error as 0.06 here training and test error decreased a little bit.
* When I did 200 iteration training error was 0.016 and had accuracy of 0.94 which yielded test error to be 0.06 from which we can conclude the train error reduced a lot, but the test error was same.
* When I further kept on increasing the iterations to 300, 400 we saw that the training error kept on increasing whereas yielding an accuracy of 0.94 on miner but the test error remains same as earlier as 0.06.

1. 

2. 

* While doing iteration for 500 we got training error as 0.016 which reduced as compared to earlier but increased the test error in the same range. Here we are able to see the model began to overfit. But by increasing boosting I found that test error is almost constant. So, we can conclude that AdaBoost is resistant to overfitting.

8) I have also estimated test error using a single decision tree, without any pruning and we got error as 8.2% which is demonstrated in the graph:-



CONCLUSION, INTERPRETATION OF RESULT AND FUTURE WORK:

The following are some (very broad) categories for the test/train stages:

*The test/train accuracy is noisier when you initially start training, but they are very strongly correlated. This indicates that you haven't quite fit the problem.

*As they both start to reduce with time, but the training error starts to decline faster than the testing error. This indicates that we're getting close to achieving a high degree of fitness.

*Eventually, the error rate of the testing set begins to rise, while the error rate of the training set continues to fall. This indicates that we have begun to overfit.

If our test accuracy is higher than our train accuracy, we're still on the training graph's far left side. There are three primary approaches to resolving this issue:

1) Utilize a method that is more suited to tiny datasets.
2) Change the model constants to better fit your training set (increasing the learning rate)
3) Obtain more data.

Hence combined classifier of 1000 decision trees far outperform on test data one consisting of only five trees, even though both perform perfectly on the training set. Indeed, extensive experiments indicate that AdaBoost generally tends to be quite resistant to overfitting.

We can also see that when we apply AdaBoost to decision tree the error rate has decreased significantly as for single decision tree test error was 8.2% and after applying AdaBoost we got test error for an average of 6.25%.

References:-

https://stats.stackexchange.com/questions/160902/is-it-possible-for-test-error-to-be-lower-than-training-error
https://geoffruddock.com/adaboost-from-scratch-in-python/
https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/