Assignment:3

Name: Uday Kumar Kamalapuram
GMU Id: G01340201
Miner User Id: uday
Public Score: 0.94
Rank: 24

AdaBoost:
AdaBoost is Adaptive Boosting algorithm, where we will apply on other learning algorithms to improve the performance. Boosting is an ensemble strategy that aims to build strong classifiers from a set of weak ones. It aims to enhance prediction power by training a series of weak models in this assignment decision stumps, each one compensating for the shortcomings of the ones before it. In this assignment, we're giving more weight to instances that are difficult to classify and less to those that are already well-classified. AdaBoost gives machine learning models more power to increase prediction accuracy.

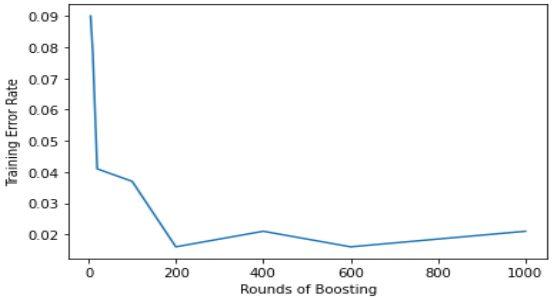
APPROACH
Selecting the Notebook:
I have chosen Google colab to start the Assignment as I am familiar with Google colab and I can work on it from the cloud.
Data Uploading and Framing:
I have imported the given text files "Assignment3-train.txt" and "Assignment3-test.txt" through files.upload() from google.colab library. I am reading the files through Pandas library read_csv() and storing them as pandas data frame with columns as there are 257 columns, In first column there is an target number either 3 or 5.
Data Pre-processing:
I have taken the target values i.e., 0^{th} column into new variable y_data and classified target value 5 as 1 and target value 3 as -1. I dropped the 0^{th} index from the given data and stored the remaining data columns into x_data.
Train-Test Split:
I have split the trained data using train_test_split from sklearn.model_selection. I have used this to get the accuracy of the model before uploading it to Miner. I have checked the accuracy on the train data by splitting the train data to x_test, x_train, y_train, y_test.
Building Model:
First, I have tried to implement the decision Stump classifier, by calculating the Gini index of each feature for that I have written function "calculateGiniForColumn" where it will calculate gini index for each column in data and we will choose minimum Gini impurity column as a decision stump as it is performing little it better than other features and I am implementing the AdaBoost algorithm on that stump. In AdaBoost first I am initializing the samle_weigths of 1/ x rows size = > 1/n. I am iterating through the number of iterations and using the decisionStumpClassifier which has minimum Gini Impurity, and we are calculating the importance of that stump by using the error rate. After that I am updating the sample_weight. I am updating sample weight for each iteration.

Following is the table for the test and train errors for different iterations.

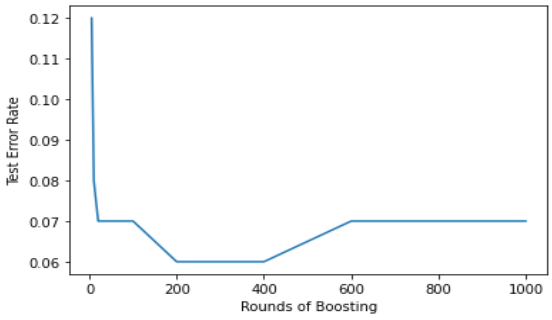
Iterations	Train Error	Test Error	Miner Accuracy
5	0.09	0.12	0.88
20	0.041	0.07	0.93
100	0.037	0.07	0.93
200	0.016	0.06	0.94
400	0.021	0.06	0.94
600	0.016	0.07	0.93
1000	0.021	0.07	0.93

Assignment:3

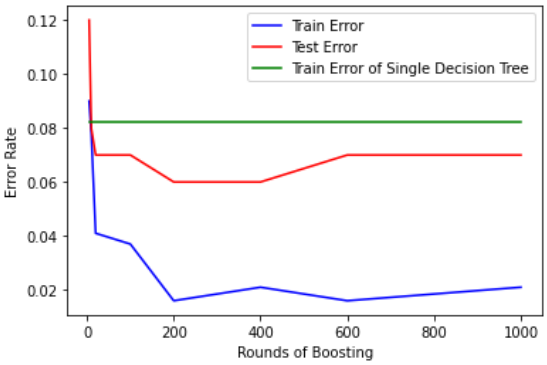
Observations:
In this assignment I have experimented with taking different Rounds of boosting and observed the test error rates.
While applying AdaBoosting on train data decision stumps, I Started with very small iterations (rounds of boosting) 5 and I Observed that the error rate on train data for 5 iterations is 0.09, whereas predicted output on test data has 0.88 accuracy In miner, therefore I got 0.12 error rate larger than train error.



When I take iterations (rounds of boosting) 20 and I Observed that the error rate on train data for 20 iterations is 0.041, whereas predicted output on test data has 0.93 accuracy In miner, therefore I got 0.07 error rate, larger than train error And here both train error and test error drastically reduced Compared to previous test errors at 5 iterations.

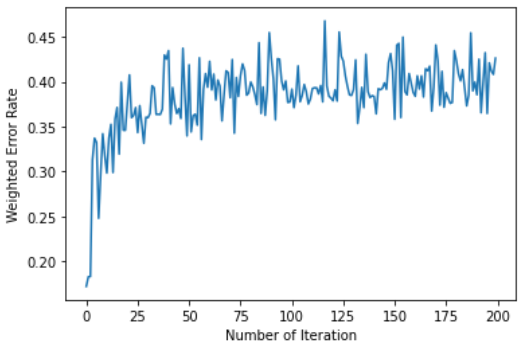


When I further Increased the iterations to 100, the error rate on train data is 0.037 there is little decrement in the error rate but in test error rate there is no change it is 0.07. When I further decreased the iterations (rounds of boosting) to 200. The train error rate got decreased to 0.16 lowest and test error rate reduced to 0.06 lowest in the train error set. This range is the good level of fit as per the observation from the graphs shown beside.

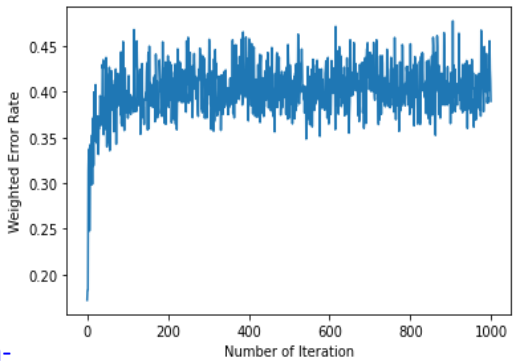


When further increase the iterations (rounds of boost) to 600. I observed that there is a decrease in the train error rate, but oppositely there is increase in test error rate, in the same range. Here we able to observe that model begin overfit. But from the further observations by increasing the iterations (rounds of boost), I observed that the test error is almost constant there is no further increase. So, we can say that AdaBoost is prone to Overfitting.

The graph to right is the weighted error rates for each iteration looping through till given iteration.



I have also estimated the test error using a single decision tree for the given data set and I got an error rate of 0.082, I plotted the line with red line in the graph. So, by applying AdaBoost on the weak learner we have achieved decrease in error rate, by applying rounds of boosting through AdaBoost.



References:

https://towardsdatascience.com/a-mathematical-explanation-of-adaboost-4b0c20ce4382

https://pandulaofficial.medium.com/implementing-cart-algorithm-from-scratch-in-python-5dd00e9d36e

https://geoffruddock.com/adaboost-from-scratch-in-python/