Machine Learning Assignment-2

Sathvik Reddy Junutula (sxj174230)

Objectives: To run SVM, Decision trees and Boosting algorithms to achieve 2 class classification to compare the algorithms that works better in two types of data sets.

Data description and reason for choosing:

1) The first dataset was provided by the instructor and it is used to apply three algorithms. The data is cleaned by removing unnecessary variables (the first two variables that had no role as variables. The variable shares are divided into two classes (high and low) based on the median value. Any other metric could also have been used but as the data is continuous, the median is used as the mean could be biases towards the high sharing values. Post the classification into factors, the data is scaled to use it for SVM and not for Decision trees to achieve the results in best and fast manner.
2) The other dataset is picked from UCI ML repository where the challenge is to identify whether the user defaults his credit card in the next month based on the previous months data. The reason for choosing this dataset is this type of problems are famous in the industry and the weight of the data is like that of the one given by the instructor (except number of variables). Also, the data is mix of continuous and categorical which makes the things a bit challenging. Again, the data is scaled except the class variable and it is then used for SVM and not for Decision trees.

Experimentation: Both the data sets were run with the three algorithms and the results were presented. The idea, process, learnings and the conclusions are provided individually for each experimentation below.

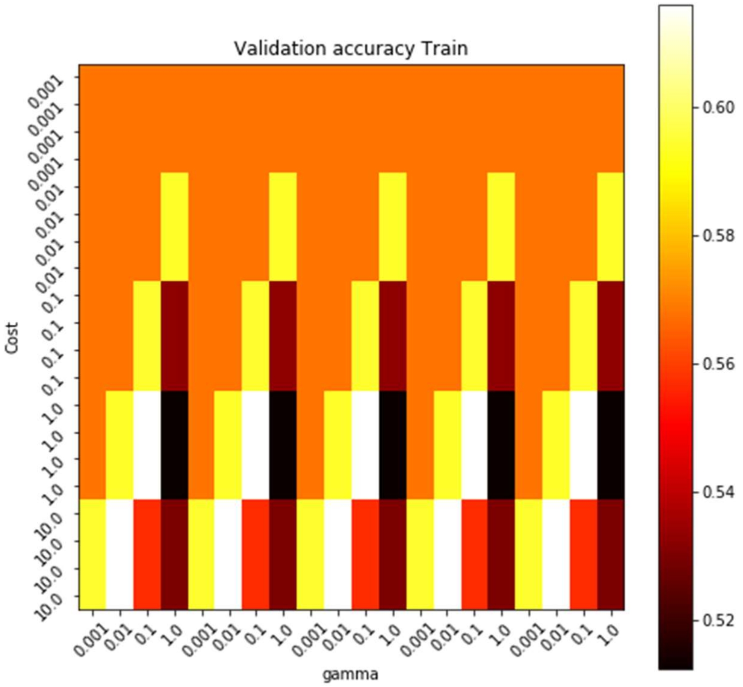
Experimentation 1: Support Vector Machines:

To run the problem three kernels were used to project the data to higher dimensions to draw the line between the two-class label's. As both the data sets have large number of variables, the use of just linear kernel might not be a great idea, hence other kernels were also used. As the polynomial kernel multiplies each variable to the rest and then the kernel is applied, it took more time than anticipated. Hence to reduce the computational cost it is ignored as of now. The other two kernel's that were used are a) sigmoid and b) radial apart from linear. The reason for choosing sigmoid and radial kernels is that they perform in a faster way when compared to polynomial according to the industry suggestions.

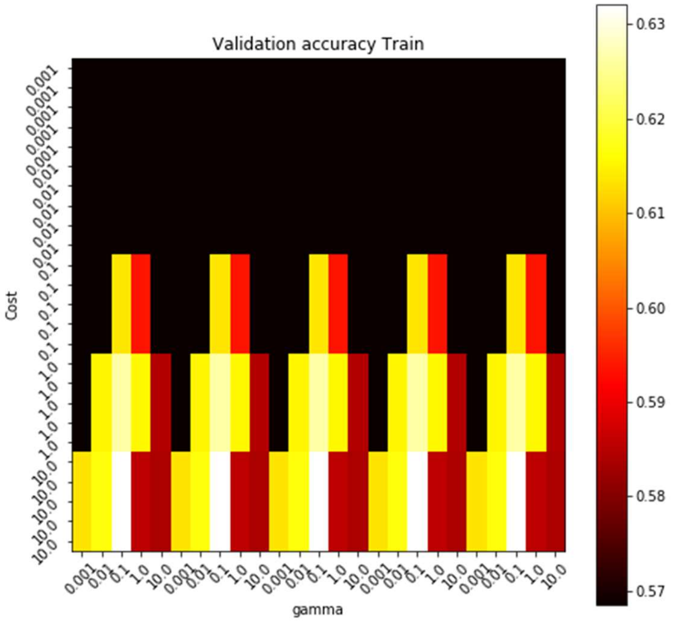
The research suggests that the results are better when polynomial kernel is used for discrete data and sigmoid for other data types. Also, the radial kernel is a kind of band pass filter and is used in a case where smoothing is necessary. Apart from these there was no hard reason to pick the kernel, but the idea was to experiment and pick the one that gives the best result for the underlying data.

First approach on dataset1:

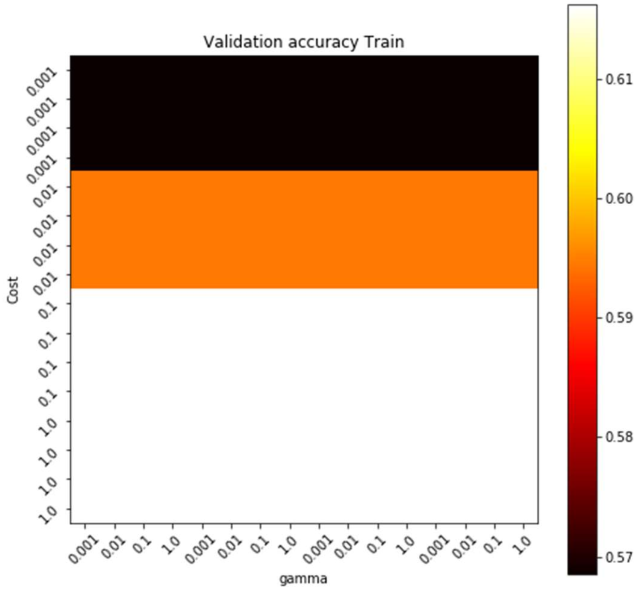
For running the SVM kernel's first the packages from R were used and as they were taking longer than usual, the implementation was later shifted to python and the results were generated in 2 ways, one for each data set. For the first data set the data is divided into 70-30 to train and test sets and 60% of the data from the train dataset is used to train the model as the calculation on the entire dataset is becoming heavy. Upon later the best parameters were selected from the results and are applied to the test set to find the accuracy (TP+TN)/(Total) to find out the accuracy. In this process the algorithm that was used was GRIDSEARCHCV, a package in python to use with additional features. In this process a set of 5 ϒ [.0001,.001,.01,.1,1] and values for 5 Cost [.0001,.001,.01,.1,1] values are passed to find out 5x5 = 25 pairs of the parameters to find the best output from the cross-validation and apply the same on the test data set for each kernel. The grid to pick the best values of ϒ and cost are presented below along with the accuracy for the model.



Grid Search plot for Sigmoid Kernel. The best parameters are found to be Cost = 10 and ϒ = 0.01



Grid Search plot for Radial Kernel. The best parameters are found to be Cost = 10 and ϒ = 0.1



Grid Search plot for Linear Kernel. The best parameters are found to be Cost = 1 and ϒ = .001

Results: For this data set OnlineNewsPopularity the least error or the best accuracy is reported for the kernel and Parameters are tabulated below:

Kernel	Cost	ϒ	Accuracy(approx.)
Linear	1	.001	62%
Radial	10	0.1	62%
Sigmoid	10	.01	65%

Results: For this data set the least error or the best accuracy is reported for the kernel sigmoid with the parameters Cost = 1 and ϒ = .001

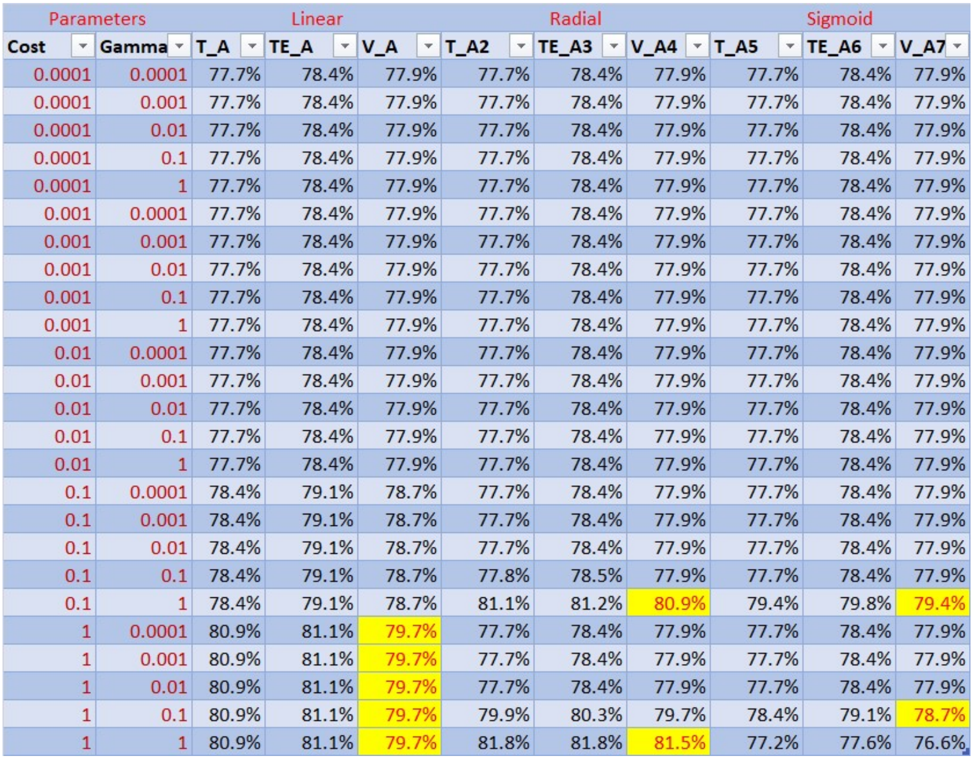
<u>Second approach on dataset2:</u>

In this approach, instead of using a grid search approach another approach (not much different) was used. In this approach

An iterative loop was used to generate 5x5 = 25 iterations to run

- Each iteration calculates the training error (on trained data), test error (test data) and Cross-validation Error (On entire data) with 3-fold cross validation for each paired value of ϒ and cost.
- Then by plotting the graphs for train, test and cross validation for the parameters ϒ and cost, a best pair of value is picked, and it is estimated to be approximation of accuracy for the future test data sets.

The results for the run along with the kernels are shown below in a tabular format and the best ones are suggested from there in a separate table.



Parameters		Linear			Radial			Sigmoid		
Cost	Gamma	T_A	TE_A	V_A	T_A2	TE_A3	V_A4	T_A5	TE_A6	V_A7
0.0001	0.0001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.0001	0.001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.0001	0.01	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.0001	0.1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.0001	1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.001	0.0001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.001	0.001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.001	0.01	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.001	0.1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.001	1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.01	0.0001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.01	0.001	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.01	0.01	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.01	0.1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.01	1	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.1	0.0001	78.4%	79.1%	78.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.1	0.001	78.4%	79.1%	78.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.1	0.01	78.4%	79.1%	78.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
0.1	0.1	78.4%	79.1%	78.7%	77.8%	78.5%	77.9%	77.7%	78.4%	77.9%
0.1	1	78.4%	79.1%	78.7%	81.1%	81.2%	80.9%	79.4%	79.8%	79.4%
1	0.0001	80.9%	81.1%	79.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
1	0.001	80.9%	81.1%	79.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
1	0.01	80.9%	81.1%	79.7%	77.7%	78.4%	77.9%	77.7%	78.4%	77.9%
1	0.1	80.9%	81.1%	79.7%	79.9%	80.3%	79.7%	78.4%	79.1%	78.7%
1	1	80.9%	81.1%	79.7%	81.8%	81.8%	81.5%	77.2%	77.6%	76.6%

Kernel	Cost	ϒ	Accuracy(approx)
Linear	1	Flexible	80%
Radial	1	1	81%
Sigmoid	0.1	1	80%

Results: For this data set the least error or the best accuracy is reported for the kernel radial with the parameters Cost and ϒ are 1 and 1 respectively.

In both the approaches the cross validation is used and implemented as there is a chance of overfitting only for the train data set to predict the approximated error using the test set which is believed to be the best approach according to the research and industry.

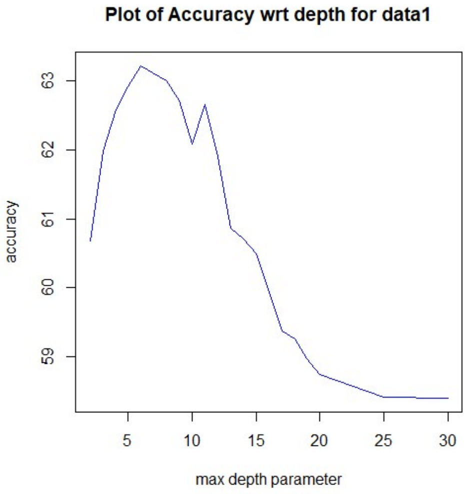
Experimentation 2:Decision Trees:

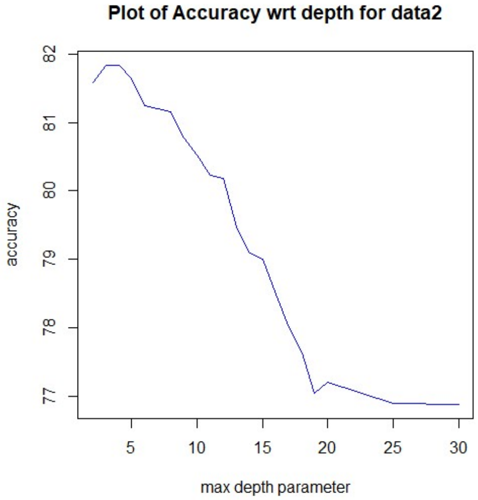
Both the datasets were classified using the decision trees by changing the tree depth, number of node, minimum number of values in the leaves bucket and changing the CP(complex parameter) which is an inbuilt parameter in rpart package. The whole optimization process is carried out on GINI index. According to the research there is not much difference between both GINI and Information Gain and the difference is not more than 2% of the use cases. Also, as Information Gain has logarithm component involved in it, the computational time is more for Information Gain. Hence GINI index is used.

CP: Also, the fit of the trees is calculated by using the complex parameter in both the runs. Any split that does not decrease the overall lack of fit by a factor of CP is not attempted. The main role of this parameter is to save computing time by pruning off splits that are obviously not worthwhile.

Essentially, the user informs the program that any split which does not improve the fit by cp will likely be pruned off by cross-validation, and that hence the program need not pursue it.

Initially each set of data set was run to find the ideal depth till which a tree can grow to yield the best accuracy. In that process a set of size c (2,3,4,5,6,8,9,10,11,12,13,14,15,16,17,18,19,20,25,30) were passed in iterations and the ideal tree depth is reported against the best accuracy. Below is the plot for two data sets for the analysis by depth.

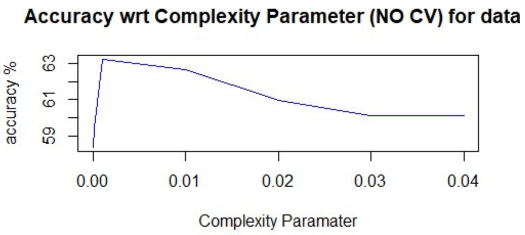


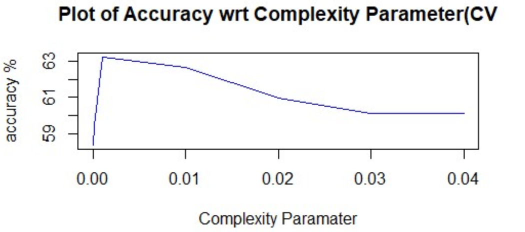


It is learned from the plots that as the depth increases for the data the accuracy drops gradually hence a tree size not more than 10 is recommended for both the datasets.

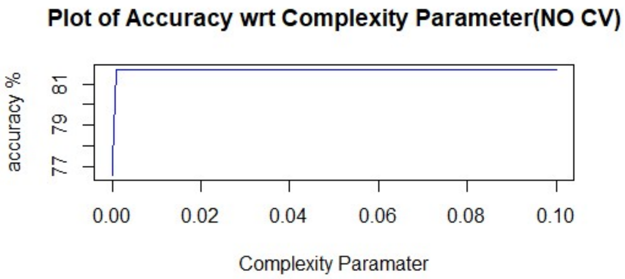
Pruning Without Cross Validation:

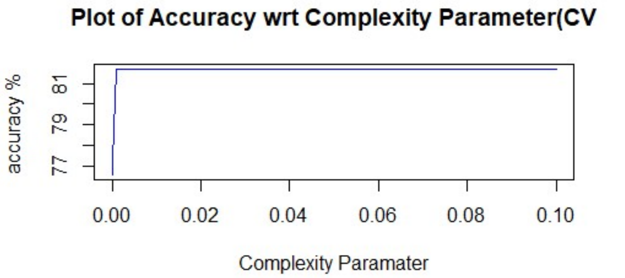
As mentioned earlier both the datasets were run subject to the cost parameter. Two cases were reported where the tree was run once with CV and other without Cross Validation to see if there are any differences to report. Each of the algorithms were passed with a set of CP c(0,.0001,.001,.01,.02,.03,.04) and the accuracies were plotted. For each iteration the trees are pruned accordingly for that value of CP and the accuracies were calculated and plotted. Below are the two graphs without and with Cross Validation for OnlineNewsPopularity data.





Below is the graph for the second dataset where the accuracy metrics were plotted against the CP parameter post pruning with the same CP parameter.





Interpretations:

a) For both data sets both the results after pruning the tree with and without cross validation yield the same result. Hence for this case Cross validation is not necessary.

b) For the first data set the ideal tree size would be 10 and the CP parameter would be as close as possible to zero but not zero. Hence a CP value of .001 can be used.

c) For the second data set the ideal tree size would be around 5-6 as the number of features are also not huge in this case. About the CP a value of .001 near to 0 can be used but not zero.

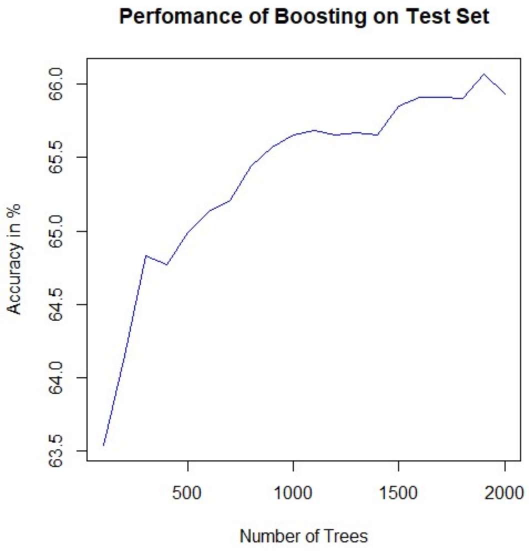
Experimentation 3: Ensemble Methods - Boosting:

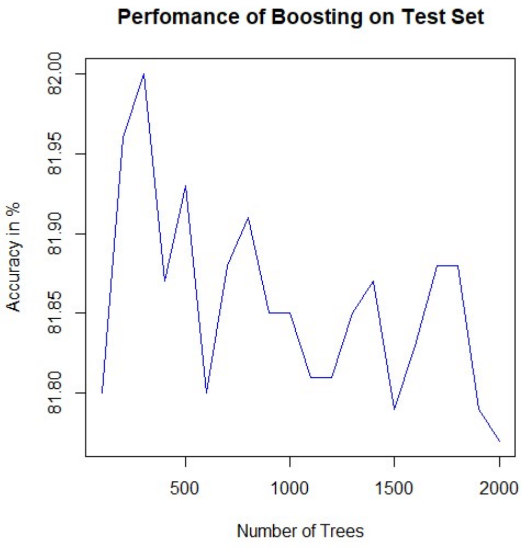
For this part of experimentation of boosting the two data sets were processed and the class labels were condensed to binary variables. For the data set of OnlineNewsPopularity the median is considered to split the data as mentioned earlier in the explanation for High and Low number of shares and for the dataset Credit the labels were already present as 1 for defaulting and 0 for not defaulting and they were unclassed and used as the package that I used requests them to be binary.

Considering the highest accuracy that we obtained for both datasets with respect to the depth parameter, both the experimentation in Decision trees are evident with high accuracy at the depth of the tree around 4-5. The parameter of interaction depth is used on the same logic and the data can shrink by the $\lambda = 0.01$. On both data sets the number of learners were limited to 2000 starting from 100 with an increment of 100 at each iteration using the for loop.

Cross Validation: A 10-fold cross validation is performed on both the datasets as the shrinkage is evident with the use of cross-validation and it avoids the overfitting of the data. Again, the predicted probabilities were reduced to 0 if it were less than 0.5 and to 1 if it were greater than 1. Post that accuracy is calculated for each iteration for both the datasets and the plots are given below:

1) Below are the graphs for the dataset OnlinewsPopularity and Credits data set for 2000 iterations against the accuracy respectively





Interpretation:

a) For the dataset OnlineNewsPopularity the dataset had a maximum test accuracy of ~66% at the learner (number of trees) at the value 2000.
b) For the dataset Credit the dataset had a maximum test accuracy of 82% at the learner (number of trees) at the value 400. The accuracy had a thresh and drought behaviour, but the variance is less and is between 80% and 82%

It is learned from both the graphs that the test accuracy kept on increasing as the learners increased. But for testing the new vector the learner number 2000 for OnlineNewsPopularity and number 400 for credits data can be used as single programme will be computationally fast and gets the results instantly.

Conclusions:

Below is the quick snapshot of all the Algorithms that were used and summarized to help pick which could be the better one along with the comments.

Algorithm	Accuracy (data1)	Accuracy(data2)	Comments
SVM	65 (sigmoid)	81 (Radial)	Time Consuming
Decision Trees	63	81	High Supervision
Ensemble(Boosting)	66	82	Intermediate

The results(accuracy) from SVM on both the datasets fall in between the Boosting and Decision Trees. So, it’s picked over the Decision trees as they report the least accuracy among the three. When the SVM and Boosting are compared, the results of Boosting are evidently better than the SVM which can be evidenced from the fact that the way the algorithm is designed to work its way from weak learning. Also, the run time and computational capacity for SVM is way higher than that of Boosting. Also, the resulting accuracy might even increase in case the number of learners is increased for Boosting (in this experiment limited to 2000). So, from the experimentation and analysis it is advised to use Boosted version of a decision tree when a binary classification problem is to be solved.