Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

Objective:

To implement the following three learning algorithms on two datasets and to perform various experiments, compare and interpret the results: -

1. Support Vector Machines (SVM)
2. Decision trees
3. Boosting

Dataset Description and Data Pre-processing:

Two datasets have been used to implement the above mentioned algorithms:-
Dataset 1: Appliances Energy Prediction dataset
Dataset 2: Absenteeism at Work dataset

Appliances Energy Prediction dataset:

- The dataset is downloaded from UCI Machine Learning repository. Here is the link to download the dataset: https://archive.ics.uci.edu/ml/datasets/Appliances+energy+prediction
- The Dataset consists of 19735 observations on 29 variables.
- None of the column contains any missing value, so no missing value imputation is required.

The target column for the Appliances Energy Prediction dataset is “Appliances”. The Appliances column here is a continuous variable containing values ranging from 10Wh to 1080Wh with e median value of 60Wh.
Since our objective is to implement the classification learning algorithms that the problem statement requires the target variable needs to be a binary categorical variable, our target variable has been converted to a binary categorical variable with a threshold of its median values 60Wh.
The column “date” has been removed from the input variables as this column is of no use for our analysis.

Absenteeism at Work dataset:

- The dataset is downloaded from UCI Machine Learning repository. Here is the link to download the dataset: https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work
- The Dataset consists of 740 observations on 21 variables.
- None of the column contains any missing value, so no missing value imputation is required.

The target column for the Absenteeism at Work dataset is “Absenteeism time in hours”. The Absenteeism time in hours column here is a continuous variable containing values ranging from 0 hours to 120 hours with a median value of 3 hours.
Since our objective is to implement the classification learning algorithms and that the problem statement requires the target variable needs to be a binary categorical variable, our target variable has been converted to a binary categorical variable with a threshold of its median values 3 hours.
The column ID has been removed from the dataset since this column is not required for analysis.

Both the datasets were partitioned into training and test datasets with split percentage of 70 as training data and 30 as testing data.

Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

Model Implementation:

Support Vector Machines (SVM):

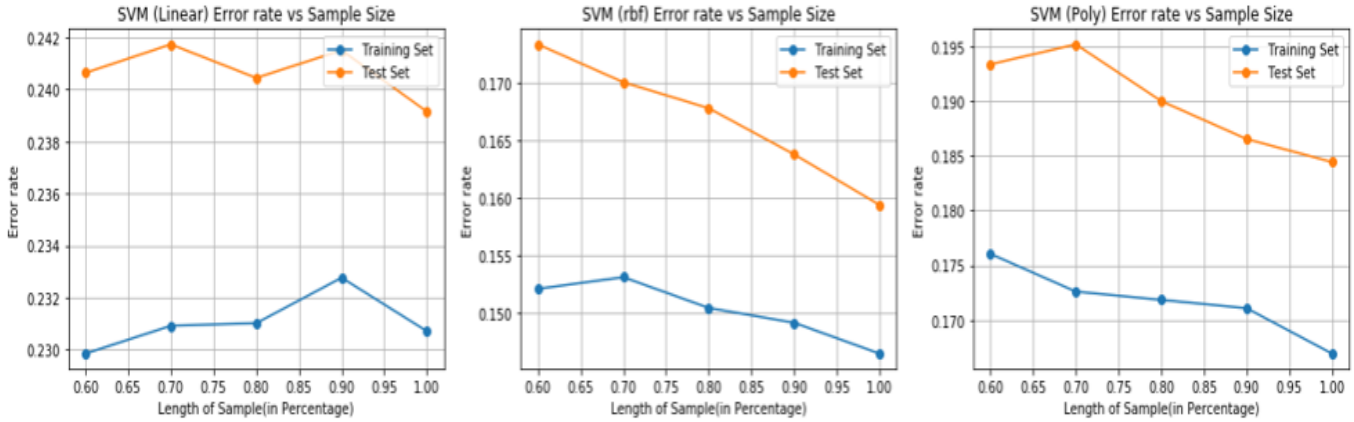
Dataset1:

The package (SVC) that is used for Support Vector Machines has been downloaded from Sci-Kit learn. SVM in this assignment has been implemented using three kernels:

1. 'Linear' 2. 'Rbf' 3.'Poly'

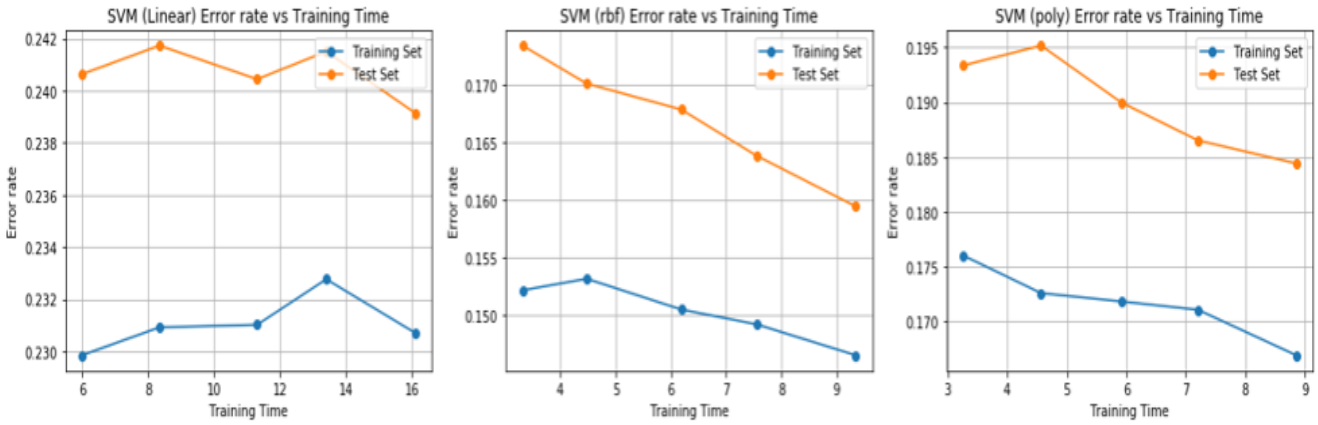
The SVM model was implemented using the above kernels on dataset 1 and the below experiments were performed. For each of the kernels models were built with datasets with different sample sizes.

Learning Curve (Error rate of different models vs Length of the sample (linear, rbf and poly):-



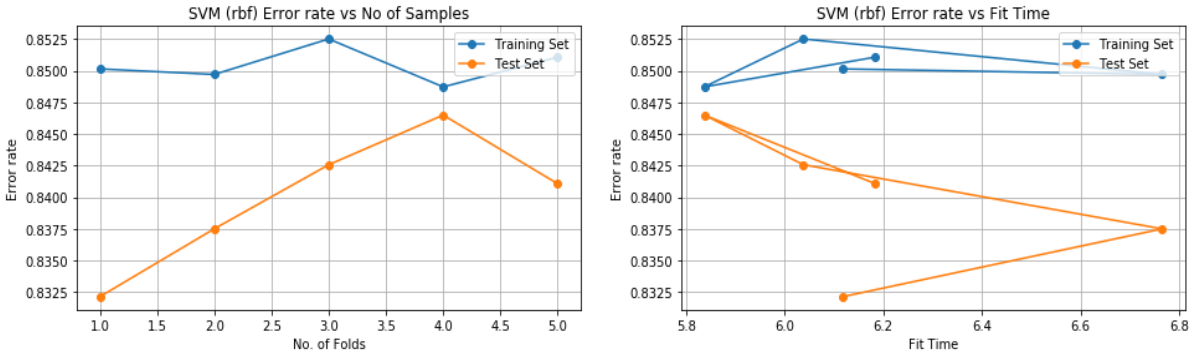
In the above graphs it can be seen that generally the error rate tends to decrease with increase in the sample size. From the above plots, we can see that SVM model with kernel as 'rbf' performs better when compared to the other kernels (Linear and Poly). It has the minimum error for the validation dataset when compared to the other models.

Learning Curve (Error Rate vs Training Time) for linear, rbf and poly:-



From the above plots for the Error Rates vs Training Time for the 'Linear', 'rbf' and 'poly' kernels for the training and test data sets, it can be seen that the error rate tends to decrease with increase in the training time of the models.

Cross-Validation: K fold cross validation was performed on SVM model that was implemented with the kernel 'rbf' using cross_validate from sklearn.model_selection package. The number of folds parameter was set to 5 and the error rates that was obtained as result was plotted as below:

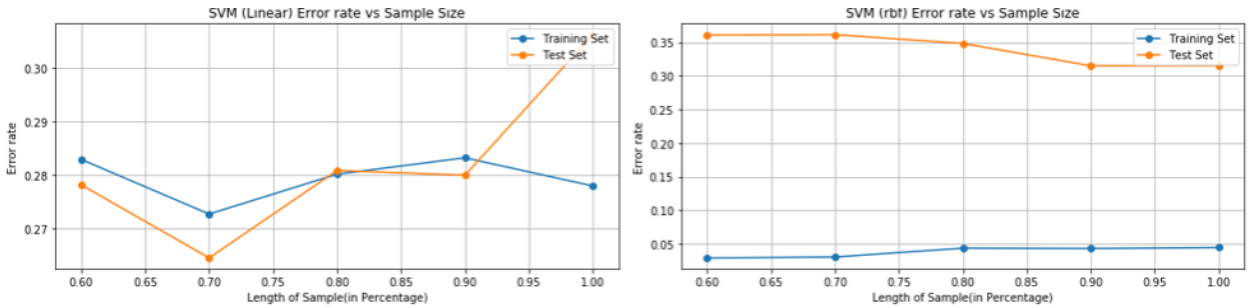


Dataset 2:

For the Adult dataset, the implementation has been done for the two kernels: 'linear' and 'rbf'.

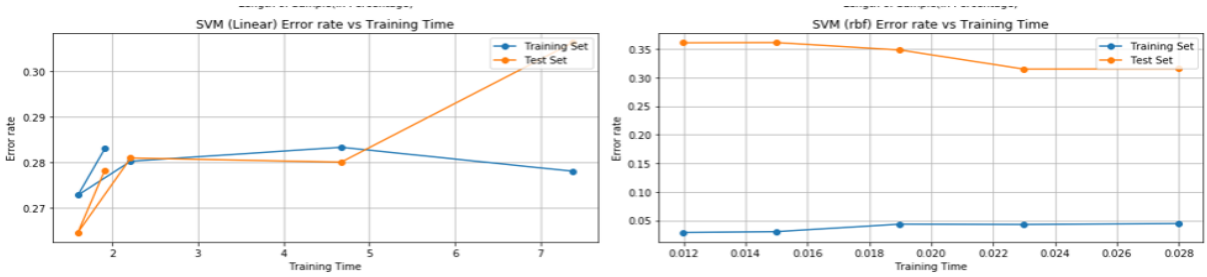
Training and building 'rbf' model for the Adult dataset took too much time to run the code. With the models that was implemented, the experiments that was conducted on dataset 1 was conducted on the dataset 2 and the following plots were obtained for the conducted experiments:-

Learning Curve (Error rate of different models vs Length of the sample (linear, rbf and poly):-



When compared to dataset 1, the error rate seems unstable for the 'linear' kernel SVM model and this can be due to some of the issues within the dataset. But for the 'rbf' kernel, the plot seems to be somewhat stable with different lengths of the sample

Learning Curve (Error Rate vs Training Time) for linear, rbf and poly: -



For training time that was captured for training different samples of the datasets was used to plot the error rate vs training time and the above plot was obtained for the training and test data sets.

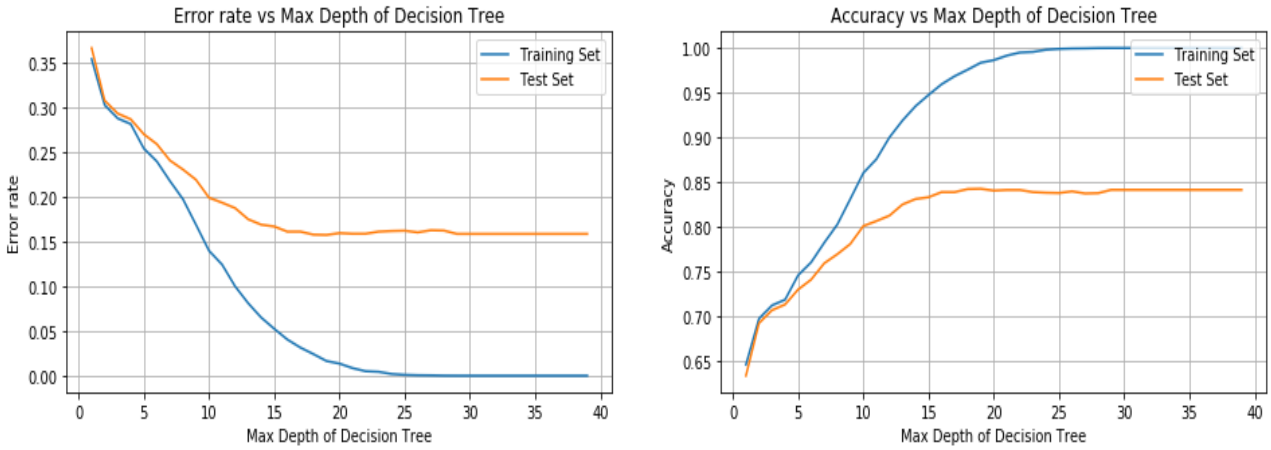
The SVM model with 'rbf' kernel seems to be performing better that the 'linear' kernel and the plot seems to be generally stable

Decision Trees

Dataset 1:

DecisionTreeClassifier was used from Sci-Kit to implement Decision Trees. For determining the Max_Depth of the decision tree model, first the model was developed for different Max_Depth and then the error rate and Accuracy was plotted against each Max Depth for the decision trees as shown in the below figure:

Learning Curve : Error rate vs Sample Size and Accuracy vs Sample Size for train and test data sets:-

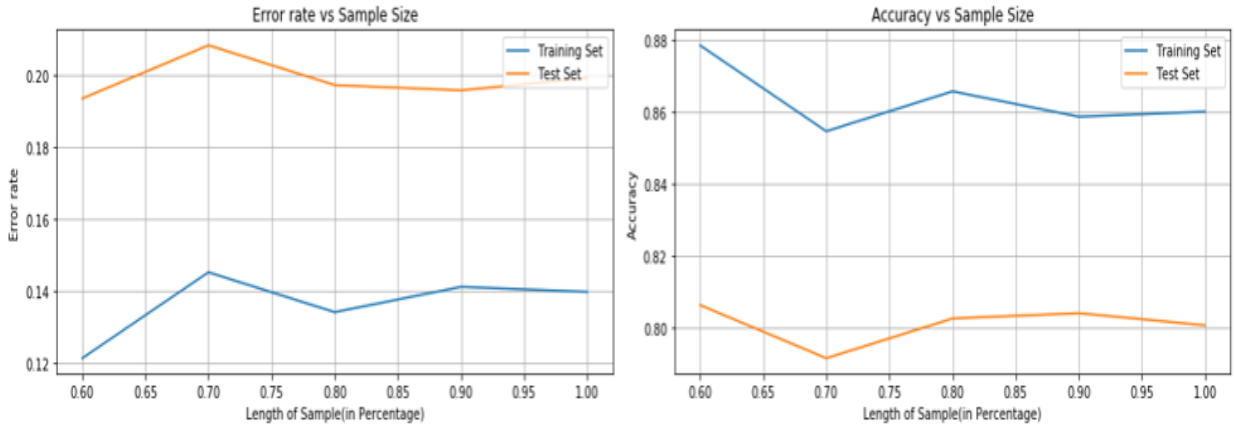


From the above plot, it can be seen that the error rate decreases with increase in the depth of the tree. For the train and test datasets it can be seen that in the above diagram, the error starts deviating from each other after the Max Depth value of 10.

Noe considering the Max Depth of the decision tree to be 10 as optimal, the Decision tree model was developed, and different learning curves were plotted as shown below:

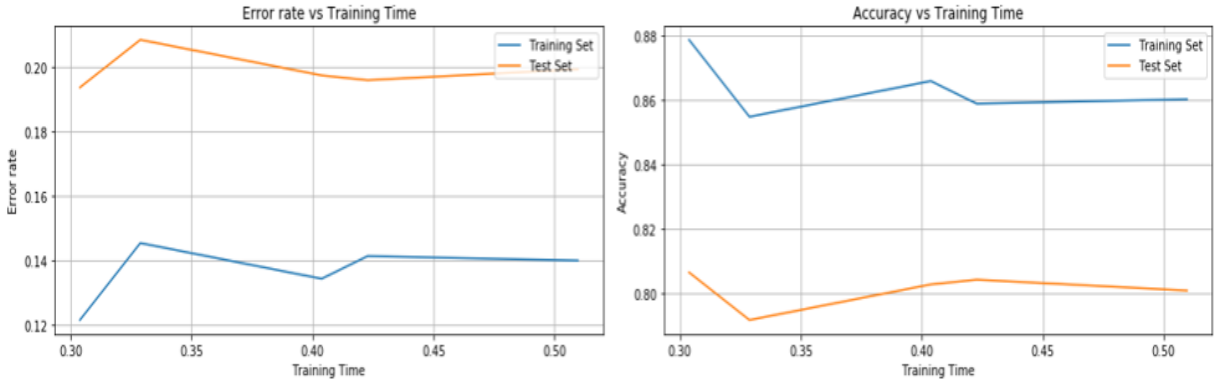
Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

Learning Curve : Error rate vs Sample Size and Accuracy vs Sample Size for train and test data sets:-



For a given MaxDepth, the above plot for Error rate vs Sample Size was obtained. The error rate seems to be slightly varying for the train and test datasets. On the right side is the Accuracy vs sample size plot.

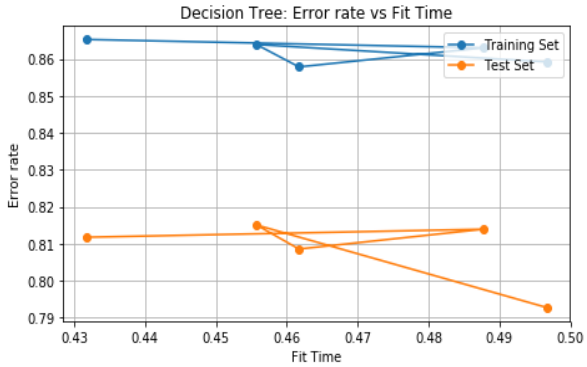
Learning Curve : Error rate vs Training Time and Accuracy vs Training Time for train and test data sets:-



The error rate usually tends to decrease in a good rate if the depth of the tree is large. The above graph Error rate vs Training Time and Accuracy vs Training for train and test sets explains the error rates of different sample train and test datasets sizes and the training time that it took for training those sample sizes.

Cross-Validation: K fold cross validation was performed on Decision Tree model that was implemented using cross_validate from sklearn.model_selection package. The number of folds parameter was set to 5 and the error rates that was obtained as result was plotted as below:

Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

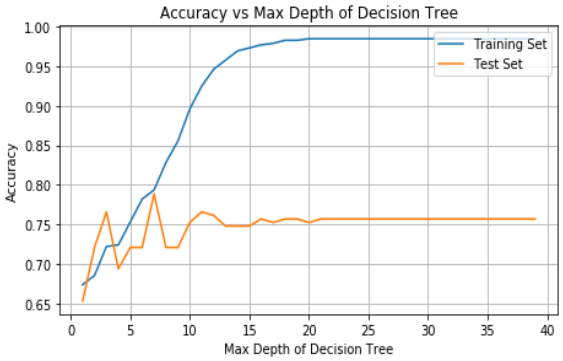


The above plot of Error rate vs No. of folds and Fit time for the train and test datasets was done for the Decision tree model.

Dataset 2

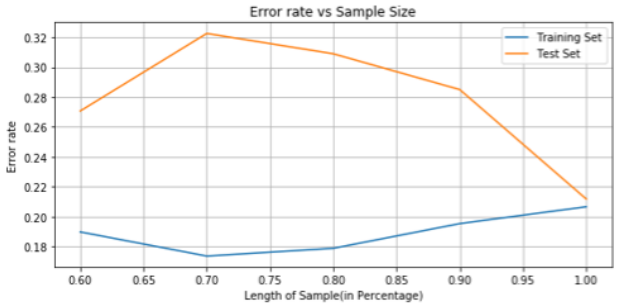
For the Adult dataset, the first experiment conducted was plotting the error rates for different depth of the decision tree and the following plots were obtained:-

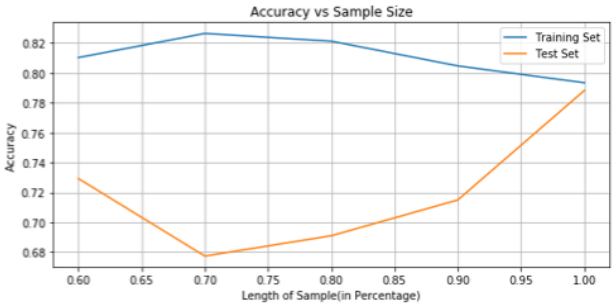
Learning Curve : Error rate vs Sample Size and Accuracy vs Sample Size for train and test data sets:-



From the above plots it can be seen that max-depth of train and test datasets starts varying at depth value of 7. So lets train a model with max_depth value of 7 and conduct various experiments on it.

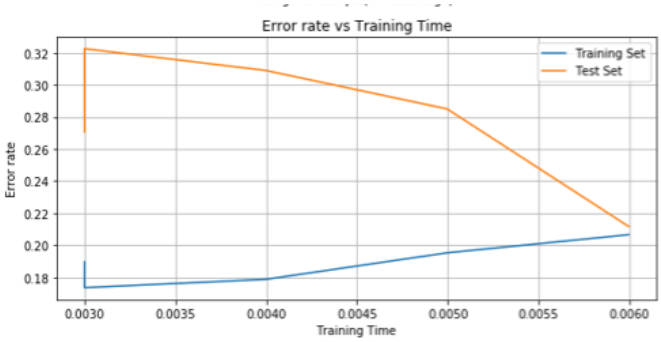
Learning Curve : Error rate vs Sample Size and Accuracy vs Sample Size for train and test data sets:-

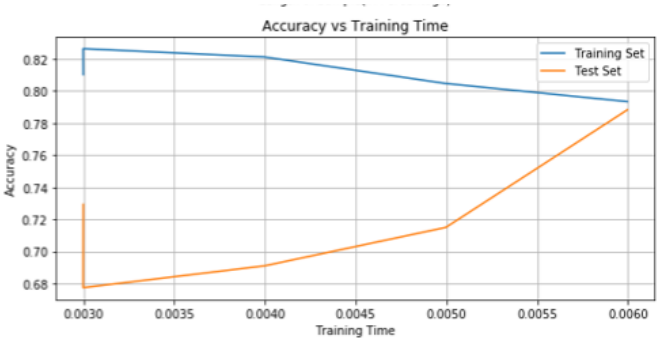




Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

Learning Curve : Error rate vs Training Time and Accuracy vs Training Time for train and test data sets:-





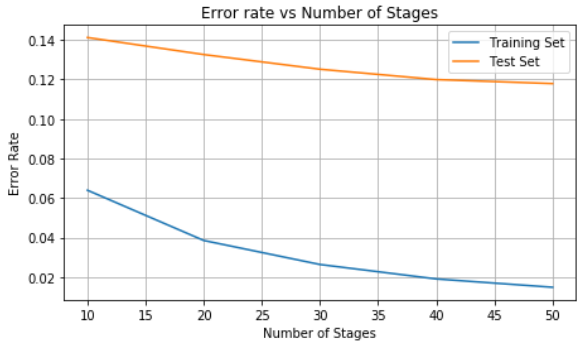
Boosting

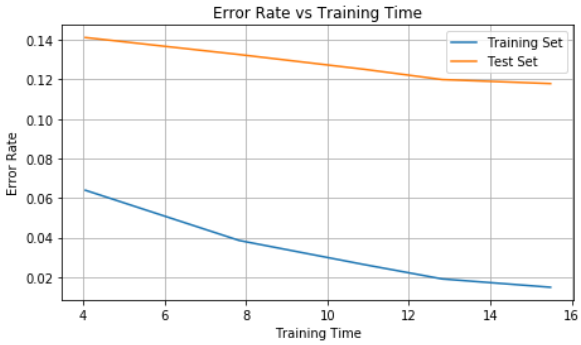
Dataset 1:

GradientBoostingClassifier from sklearn.ensemble was used to implement Boosting classifier. Consider the max_depth of 10 to be good (from experiment perspective) for the given dataset using the Decision Tree model which was implemented earlier.

The first experiment here that was performed was to determine the error rate for different stages of boosting. The training time for training up the model for different stages of boosting was also determined and the below plots were obtained:-

Learning Curve : Error rate vs No. of boosting stages and Accuracy vs No. of boosting stages for train and test data sets:-



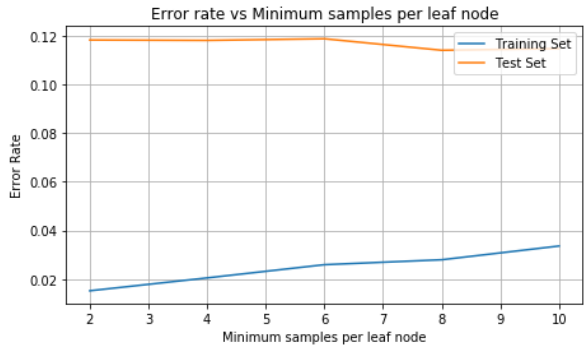


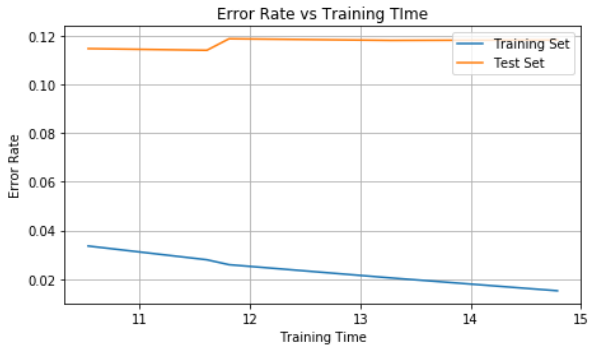
From the above plot for Error rate vs Number of boosting stages, it can be seen that the error rate tends to decrease for the Appliances dataset with increase in the number of boosting stages for both the train and test datasets.

Also the from the above plot for Error rate vs Training Time of each boosting stages, it can be seen that the error rate tends to decrease as there is increase in the training time of the models with each boosting stages

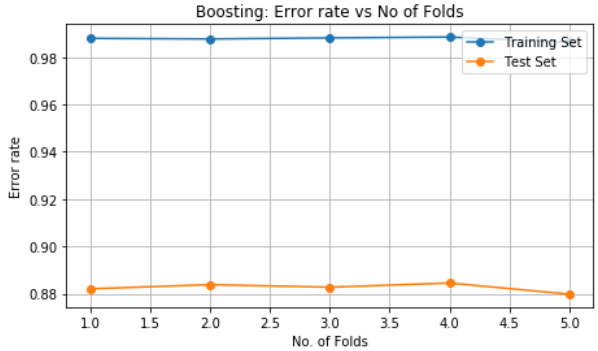
Another experiment that was performed was pruning for the model that was developed here for boosting. From the above experiment, it can be understood that the Error rate is minimum for the number of boosting stages is 50. Now, by fixing the number of boosting stages at 50, Error rate was determined for different values of Minimum sample per leaf node. Below is the plots for the error rate vs minimum samples per leaf node.

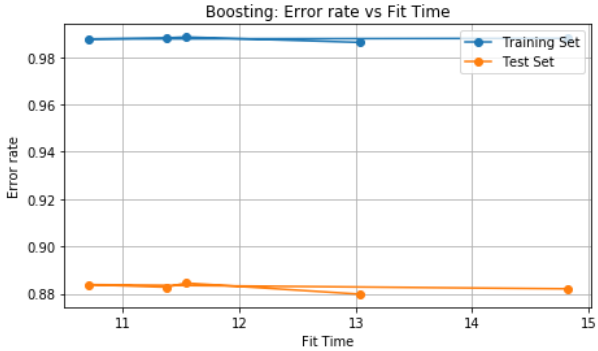
Learning Curve : Error rate vs Minimum samples per leaf node and Error Rate vs Training Time for train and test data sets:-





Cross-Validation: K fold cross validation was performed on boosting that was implemented using cross_validate from sklearn.model_selection package. The number of folds parameter was set to 5 and the error rates that was obtained as result was plotted as below:





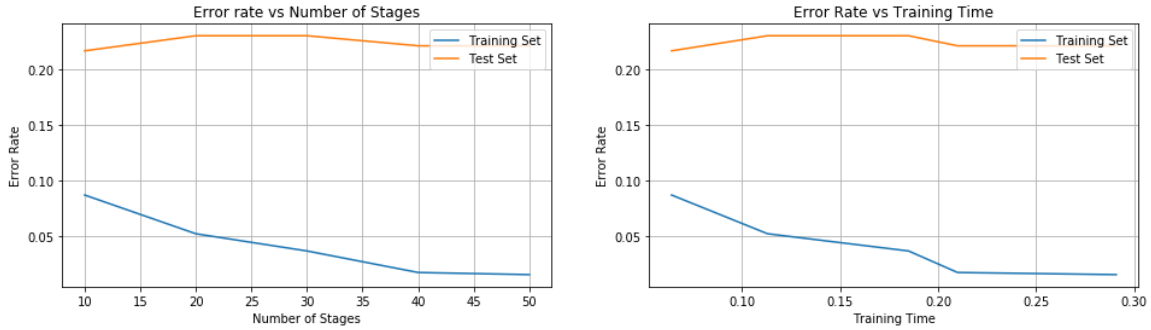
Dataset 2:

For the Adult data set consider the max_depth of 7 to be good (from experiments perspective) for the given dataset using the Decision Tree model which was implemented earlier.

The first experiment here that was performed was to determine the error rate for different stages of boosting. The training time for training up the model for different stages of boosting was also determined and the below plots were obtained:-

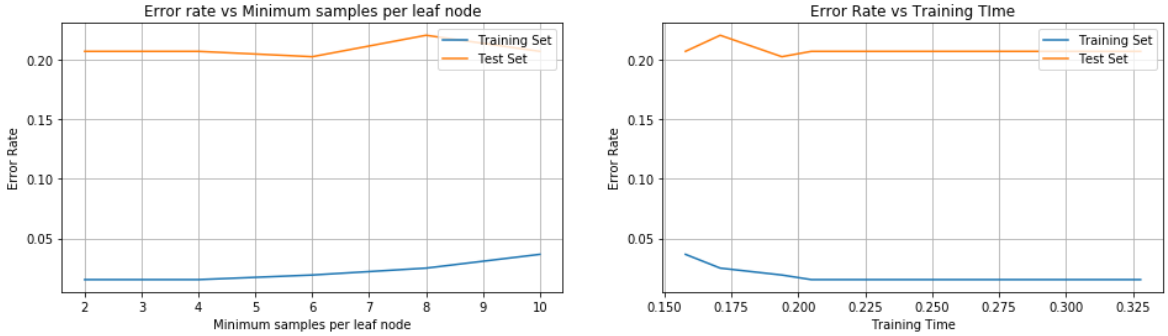
Bharath Kumar Natarajan (BXN180005)
BUAN 6341.003 – Applied Machine Learning (Assignment 2)

Learning Curve : Error rate vs No. of boosting stages and Accuracy vs No. of boosting stages for train and test data sets:-



In the above plots, the error rates of the train dataset tends to decrease with increase in the number of stages and the training time, but the error rates of the validation dataset is not that much decreasing with the number of stages and the training time.

Learning Curve : Error rate vs Minimum samples per leaf node and Error Rate vs Training Time for train and test data sets:-



Conclusion:

Best Model in DataSet1 : The best algorithm in dataset 1 as per the implementation is Boosting with an Training Accuracy of 98.5 percentage. For this dataset worked well giving a lower error rate when compared with the other algorithms that were implemented in this assignment such as Decision tree and SVM models

Best Model in DataSet1 : Bes The best algorithm in dataset 1 as per the implementation is Boosting with an Training Accuracy of 98.456 percentage. For this dataset worked well giving a lower error rate when compared with the other algorithms that were implemented in this assignment such as Decision tree and SVM models

Comparing the dataset1 and dataset2 , dataset1 gives more clearer results than dataset2. This maybe due to the consistency of data in dataset1 when compared to dataset2 and also the collinearity of the input variables in dataset2 may be high when compared to dataset 1.

In Practical there are no models that are perfect. It depends on the dataset that we are choosing for the analysis.