Report

This report depicts the implementation of three supervised learning methods namely, Support Vector Machine (SVM), Decision Tree, and Boosting on two distinct data sets. The goal is to analyze and understand the behavior of these supervised learning methods on the datasets. The algorithms are performed with the help of Sckitlearn library in Python language (on Jupyter notebook).

Dataset: Facebook Comment Volume

For this data, all the model implementations throughout this report are performed on the dataset variant 3. The dataset is same as taken in assignment 1 and you can find the details about the dataset here. Since the dataset is familiar, I am going to move ahead and directly show the implementations performed using this dataset. The dataset is not a classification one and hence following are the preliminary steps performed on this dataset before passing the data through models:

1. Converting the Target variable into a classification problem with values 0's and 1's.
2. I have treated Target variable as 0 if there are no comments and 1 if there are any comments
3. Splitting the dataset into 70:30 ratio
4. Performing feature scaling on continuous features
5. Training the dataset on different supervised models for good outputs

After converting the Target variable into categorical, I found that 55% of these values are 0's and the rest are 1's. Due to this distribution, I found it okay to go ahead with the implementation and set the threshold to 0. Moreover, taking median or mean as a deciding threshold would be misleading since the data is highly skewed towards right. Hence, the reason for selecting threshold as 0.

Further, I have selected 17 features out of all; based on the results obtained in Assignment 1. One more reason of choosing less features (and significant ones) is to avoid the computation power and time during the executions. The chosen variables are namely, 'Likes', Checkin, PageTheme, CC1, CC2, CC3, CC4, PostLength, PostShareCount, HLocal, PostPublishSun, PostPublishedMon, PostPublishedTue, PostPublishedWed, PostPublishedThu, PostPublishedFri, PostPublishedSat and a TargetClass as my label.

Support Vector Machine

Throughout this report for both datasets, I have performed various kernels on the dataset during the implementation and have selected three best ones based on their results.

1. Linear Model:

Due to the large volume of training dataset, cross validation was not performed for Linear SVM. The model was trained for different values of C [0.0001, 0.1, 1] and the best accuracy for the test data was obtained at C = 1. Please go through the following Table for measures.

2. Polynomial Kernel:

I have implemented this kernel over different degree of polynomial and plotted the learning curve of

Linear SVM Kernel	C = 0.0001	C = 0.1	C = 1
Test accuracy	0.6894	0.7999	0.8004
Confusion matrix	[[19519 718] [10565 5528]]	[[18958 1279] [6021 10072]]	[[18912 1325] [5925 10168]]

Table 1_0 Linear SVM performance for different C values

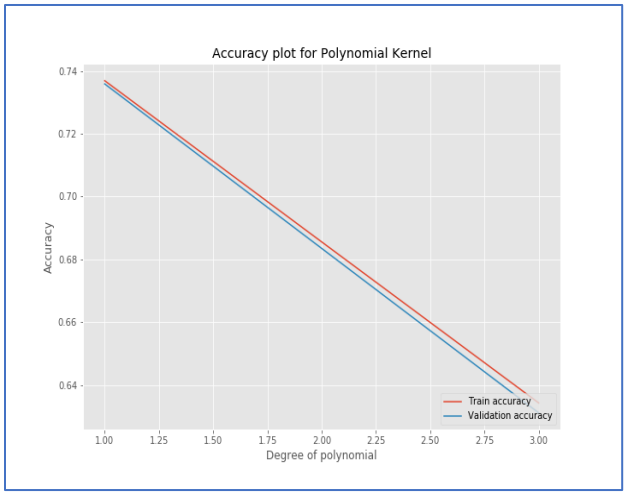

Figure 2_0 Train and Validation accuracy plot

Figure 2_1 Train, Validation and Test accuracy plot

training and test data set. I have performed cross validation with k = 3-fold (to reduce the execution time). From the Figure 2_0, it is clear that the mean accuracy of both the training and validation dataset decreases with increase in the degree of the kernel function. For this dataset, the model with degree 2 gives the best accuracy and generalizes it well. The accuracy obtained is 75.13% with confusion matrix [[19350 887] [8148 7945]].

3. Sigmoid Kernel

Sigmoid Kernel behaves like logistic model and will be of help in classifying the output labels. I have implemented this kernel for different C values [0.0001, 0.1, 1] and plotted the corresponding train and test accuracy for 3-fold cross validation. It is seen that there is not much of difference in the accuracy of the train and validation set. This model performs low on the data set with accuracy of 71.65% and confusion matrix [[15019 5218] [5078 11015]]

Decision Tree

Decision tree is another algorithm which helps to overcome the amount of time taken by SVM for computing the values. Training a data till the tree grows to its extreme may result in overfitting the dataset. In order to avoid this, pruning is performed on the data during fitting the model. Due to pruning, the model has less variance added and it can perform better when subjected to new unknown values.

I have implemented the entropy approach for training the data by pruning the trees with tree depth as 3, 5 and 10. Figure 2_0 shows linear increase in accuracy of the train dataset as depth increases while test and validation accuracy increases slowly after the depth 5. The best accuracy is obtained for the tree with max depth = 10 showing overall test accuracy of 81.52% and confusion matrix is [[17350 2887]
[3823 12270]]

AdaBoost Algorithm

This algorithm uses base learner as Decision Tree classifier and if we don't prune the dataset, the model will overfit the data and may result in 100% accuracy. In order to avoid this kind of behavior and make your model more realistic that can handle unknown values properly, we perform pruning. Pruning is done with three values 3, 5 and 10. The best results obtained are for the model with pruned value as 5. Following table gives the essential measures. The below figure shows that the as the depth increases the testing accuracy degrades but the training accuracy increases. This is a case of overfitting the model.

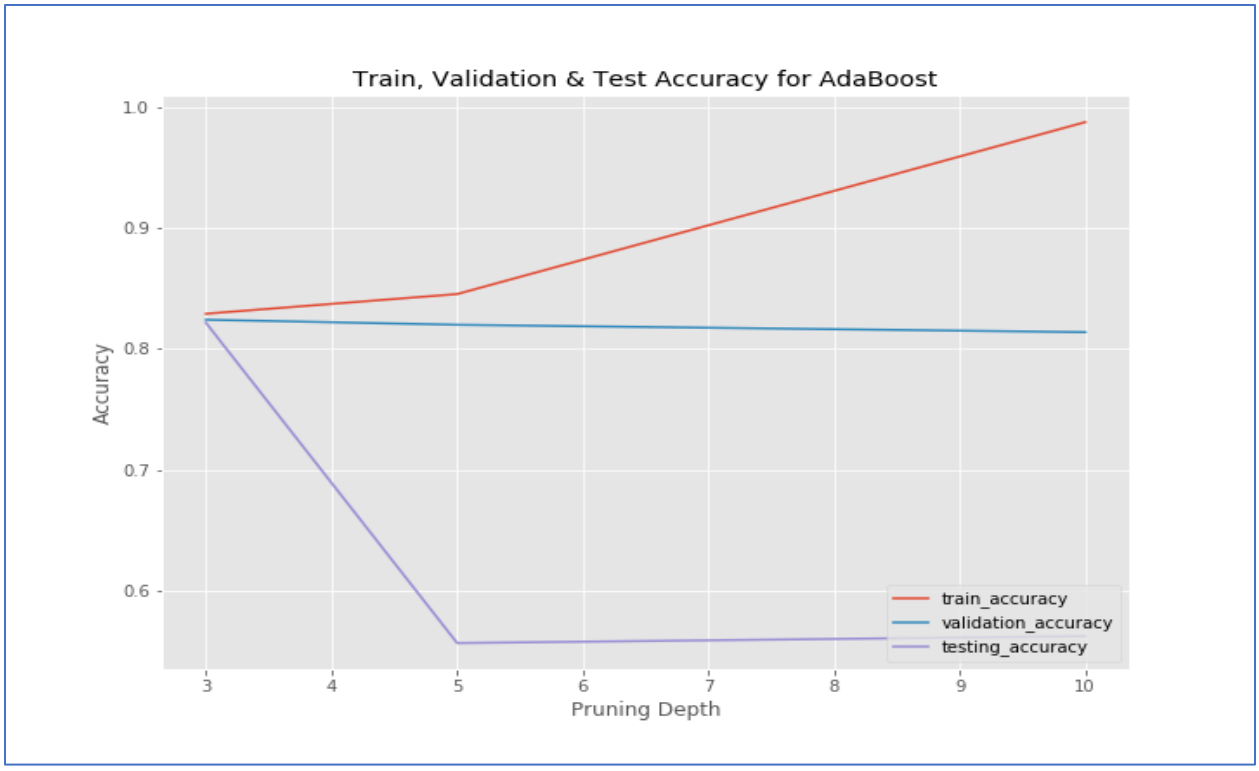

Figure 3_0. Train, Validation & Test accuracy for AdaBoost

Overall Performance Comparison

The figure below shows the accuracy of the different models that gave best results on the test data. As you can figure out, Decision tree has shown the best accuracy followed by Linear SVM. For dataset 2, the best model is Decision tree with pruning of depth 10.

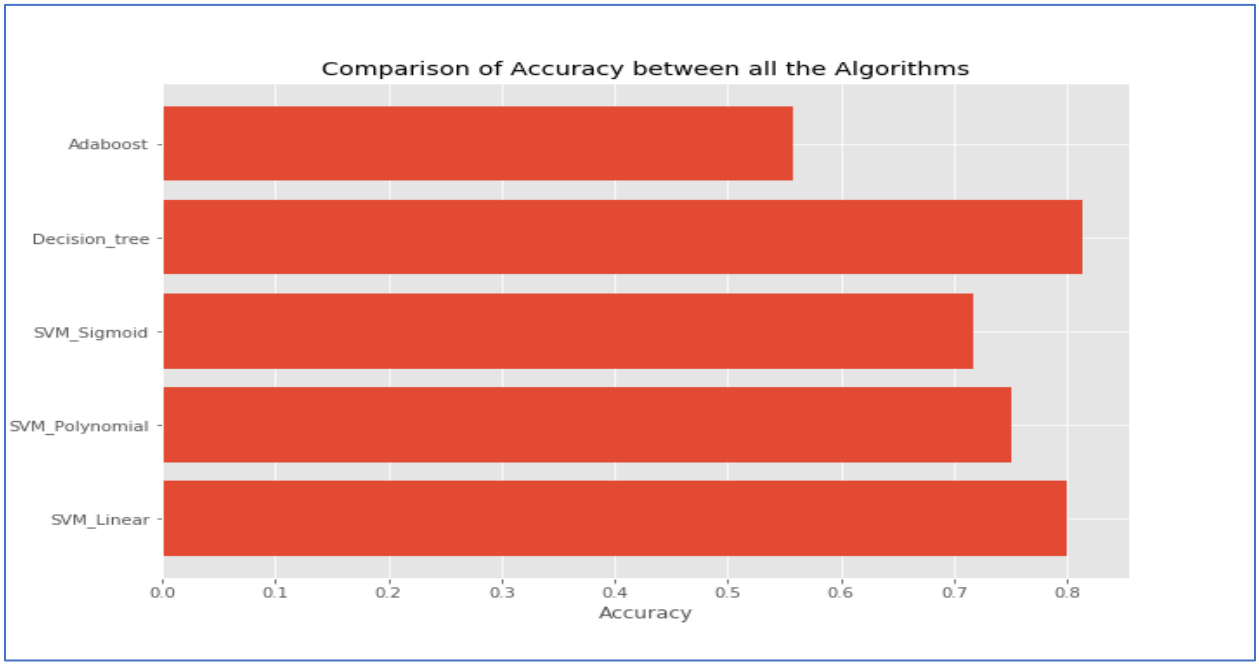

Figure 3_1. Overall Performance plot

Learning Curve – Decision Tree

The learning curve is plotted for the best performing model i.e. Decision tree in case of dataset 1. As you can see, the difference between training and cross-validation is reducing as expected. With increase in the sample sizes at each split, the error reduces for the training dataset. Likewise, when new samples are exposed to the model, it classifies them slowly with increasing sample size and the error eventually becomes constant.

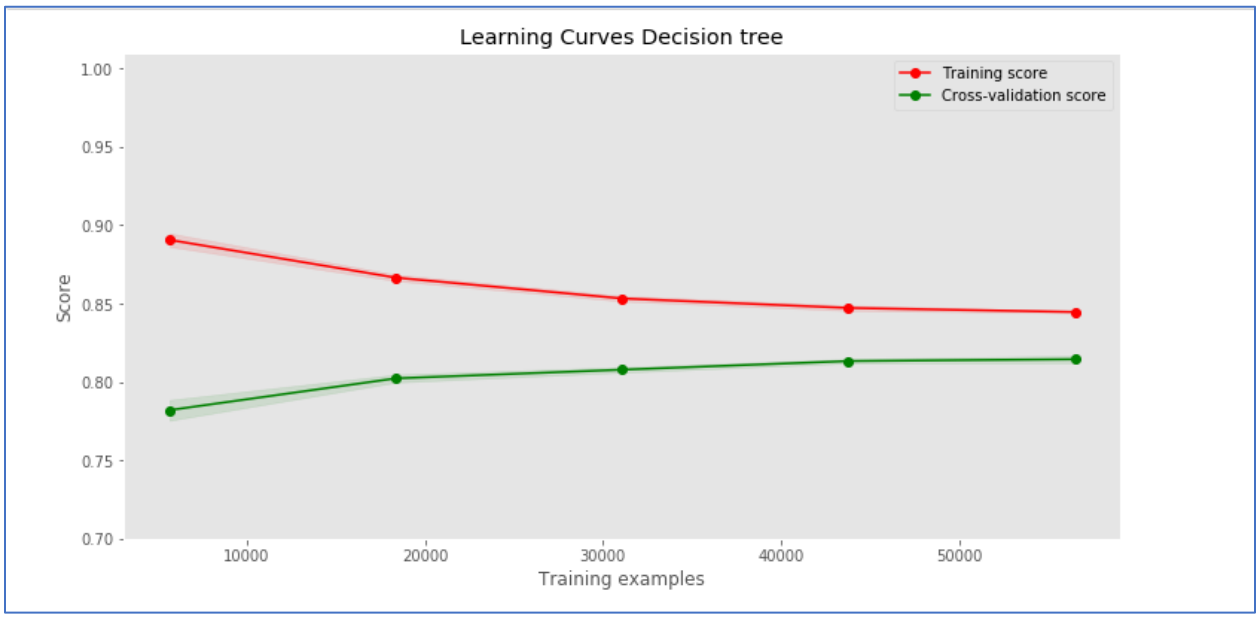

Figure 4_0. Learning Curve – Decision Tree

Conclusion

We can conclude that for Dataset1; the best accuracy is obtained by Decision tree and that for Dataset 2 it is obtained from the RBF Kernel. This assignment has made me think on datasets from a broader aspect regarding data handling, feature selection and thinking from different point of views. Overall, it was a great practice to work on this report. Both the datasets were very diverse in terms of the information provided and it was really challenging to work on both datasets.

Additional things that might have resulted in better accuracy could be; performing Principal Component analysis on the data and reducing the dimensions. Picking only those components which covers most of the variance in the dataset.