BUAN 6341 APPLIED MACHINE LEARNING ASSIGNMENT 2

Project Report

Support Vector Machines (SVM), Decision trees and Boosting

By Animesh Kansal (axk169531)

Datasets Used:

1. Online news popularity data set (39644, 61)
2. Telecom Churn data set (7043, 21)

Scripting Language: Python as scripting language

Tasks:

1. You need to explain why you think this data set and the corresponding classification problem is interesting. Divide your data sets in train and test sets.

Telecom Churn is a typical Classification data set. It has 21 features and 7043 observations which is ideal for Classification algorithms like SVM and decision tree. It has 3 Numerical features, rest all are categorical features except customer ID, which we dropped as a first thing. Then we divided the data set as this : from sklearn.model_selection import train_test_split and test_size=0.3

2. Download and use any support vector machines package to classify your classification problems. Do it in such a way that you are able to easily change kernel functions. Experiment with at least two kernel functions (in addition to linear) of your choice. You can pick any kernels you like (shown in the class or not).

I used **sklearn package and used these kernels Linear, RBF and sigmoid Kernels**

3. Download and use any decision trees package to classify your classification problems. Experiment with pruning. You can use information gain or GINI index or any other metric to split on variables. Just be clear to explain why you used the metric that you used.

Scikit-learn in python uses ‘gini’ as default.

I tried changing: gini to entropy, but it doesn’t appear to matter

On further investigation, I found out :

- Gini is intended for continuous attributes, and Entropy for attributes that occur in classes (e.g. colors)
- "Gini" will tend to find the largest class, and "entropy" tends to find groups of classes that make up ~50% of the data
- "Gini" to minimize misclassification
- "Entropy" for exploratory analysis
- Some studies show this doesn't matter – these differ less than 2% of the time
- Entropy may be a little slower to compute

4. Implement (or download) a package to use a boosted version of your decision trees. Again, experiment with pruning.

I used **from sklearn.ensemble import AdaBoostClassifier**

Telecom Churn Analysis Dataset

Algorithm	Support Vector Machine	Decision Tree	Decision Tree with Boosting
Train Set Accuracy	77.08%(Linear) 91.37%(RBF) 73.3%(Sigmoid)	99.837	99.837
Test Set Accuracy	75.97%(Linear) 77.35%(RBF) 73.7%(Sigmoid)	69.763	72.654
How to overcome overfitting/ improve model	Optimizing C and gamma values	Pruning, depth = 4	Pruning, depth = 1
Train set Accuracy after improvements	77.06%(RBF) 79.99%(Sigmoid)	79.622	81.119
Test set Accuracy after improvements	76.73%(RBF) 79.15%(Sigmoid)	79.384	79.143

In telecom churn dataset, SVM earlier was showing average of 75% error and Decision tree was showing 69% before applying any improvements in that model.

Later in SVM, when I optimized c and gamma values , and also Decision tree was pruned , lead to improvement in accuracy of both , SVM increased from 75% (avg.) to 78%(avg.) whereas DT improved from 69% to 79% . DT with Adaboost also increased from 72% to 79.143

In this case, I will choose DT with Adaboost as the best model for the Telecom project, as the depth of the resulting model is 1, hence this is simplest and has Test accuracy is almost equivalent to max among all the three.

Online News Sharing Dataset

Algorithm	Support Vector Machine	Decision Tree	Boosting
Train Set Accuracy	50.5%(Linear) 53.5% (Sigmoid)	100.000	100.000
Test Set Accuracy	49.67%(Linear) 53.04% (Sigmoid)	57.685	57.945
How to overcome o verfitting/ improve model	Optimizing C and gamma values but it was very slow. used PCA , because num ber of features are to o much, no significa nt change	Pruning, depth = 7	Pruning, depth = 1
Train set Accuracy after improvements	50.58%	67.286	68.177
Test set Accuracy after improvements	51.1%	64.175	64.810

In telecom online news popularity dataset, SVM earlier was showing average of 51% error and Decision tree was showing 57% before applying any improvements in that model.

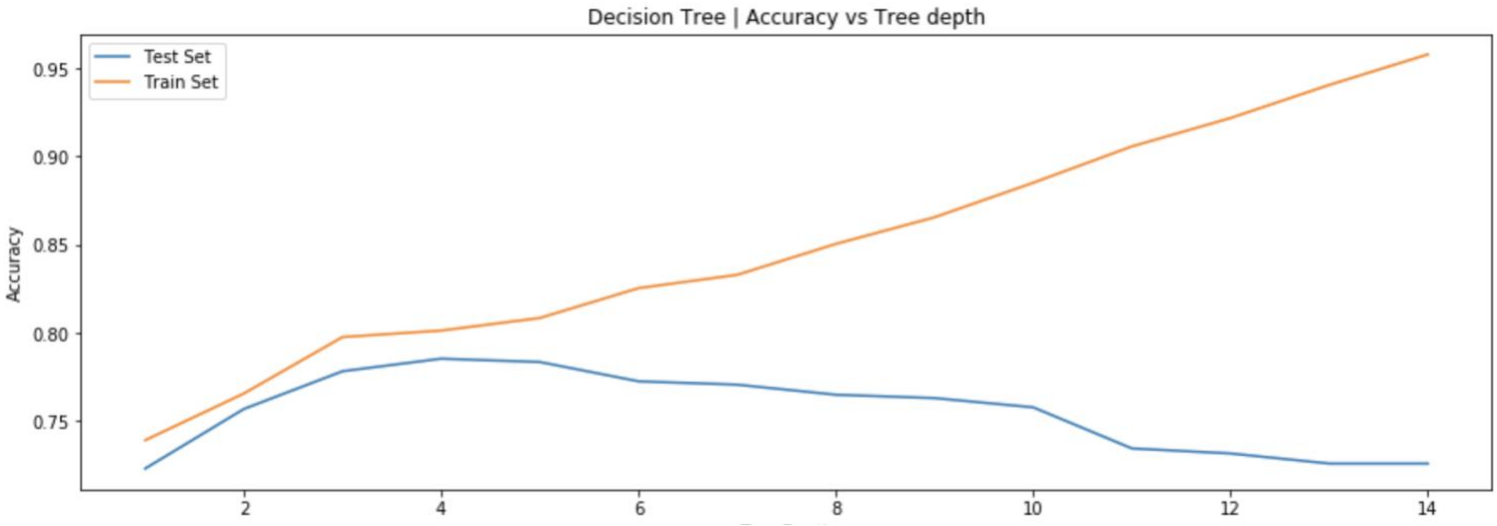
Later in SVM, when I optimized c and gamma values, the process was very slow because too many dimensions, so used PCA, because number of features are too much, but still no significant improvement happened. whereas Decision tree was pruned, leading to improvement in accuracy, DT improved from 57% to 67%. DT with Adaboost also increased from 57% to 65%

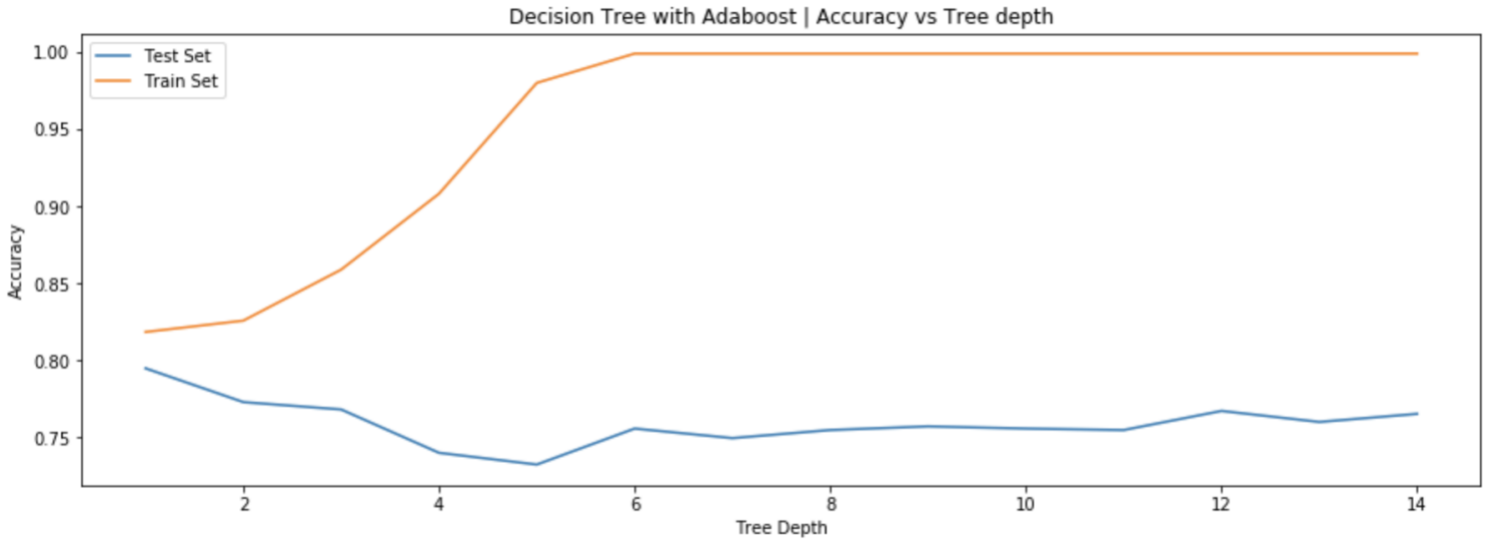
In this case, I will choose DT with Adaboost as the best model for the Telecom project, as the depth of the resulting model is 1, hence this is simplest and has least Test error among all the three

Both the datasets have shown almost similar graphs for these:

1. **Decision Tree** - > Accuracy vs Tree Depth
2. **Decision Tree with Adaboost** -> Accuracy vs Tree Depth
3. **Learning Curve** -> Error vs Sample Size

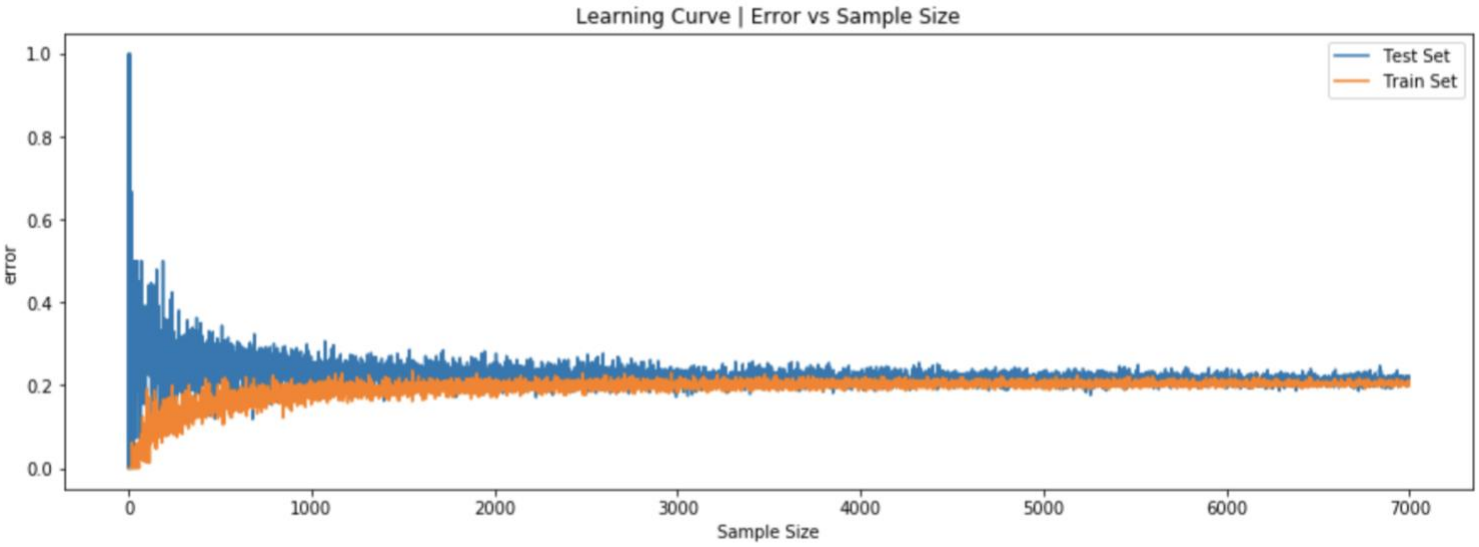
As shown below :



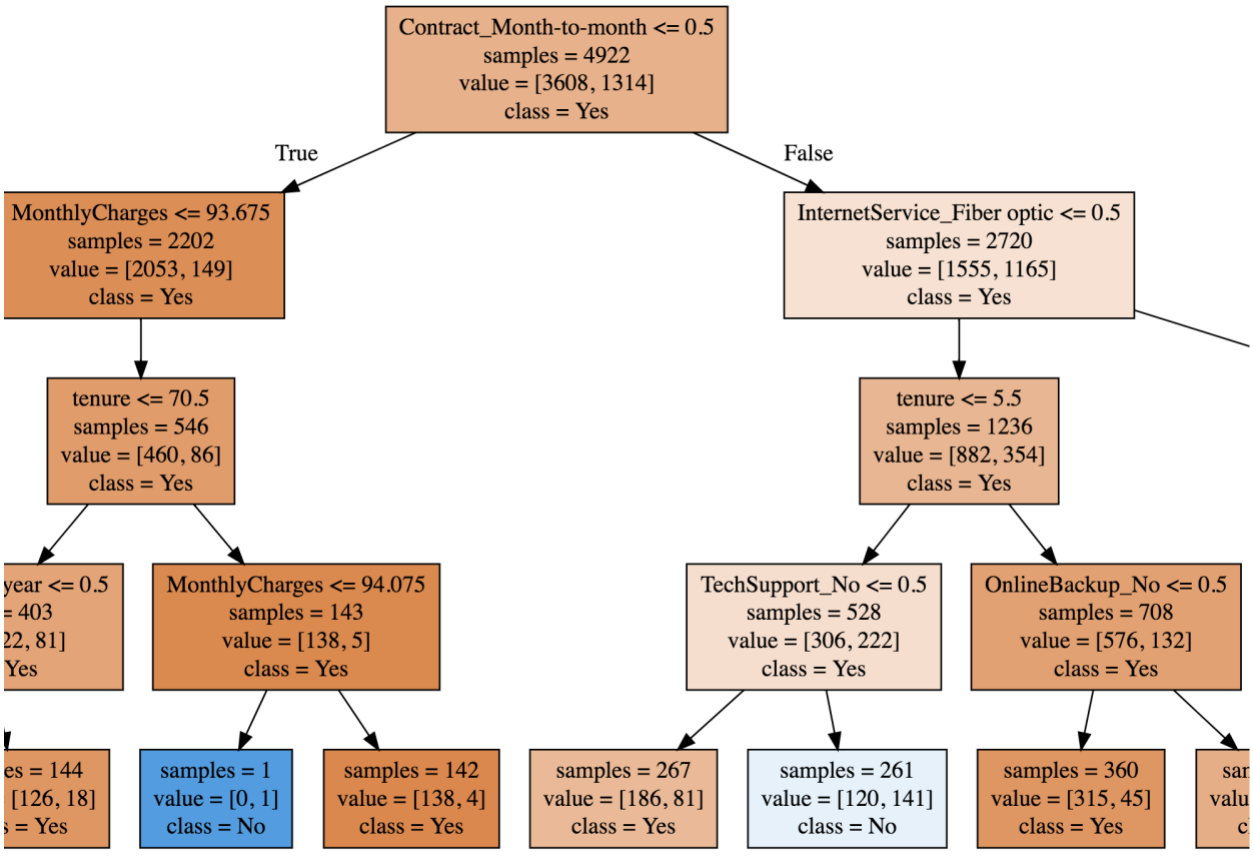


How did you do pruning and when and why did you decide to stop?

I draw the graphs of tree Depth vs Accuracy and can clearly visualize that in graph 1, to prune the tree at depth = 4



Decision tree drawn for Telecom churn prediction, but we can easily view it completely in python jupyter notebook



What additional things can you do?

Additional things we can do is:

1) Cross validation
2) Dimensionality reduction

If yes then why didn't you implement it?

Yes, Cross Validation would help definitely, I tried implementing but it was not creating significant difference, so removed it. Maybe I was implementing it wrongly. In interest of time frame, I could not do. But I have a strong feeling that CV will definitely increast test accuracy.

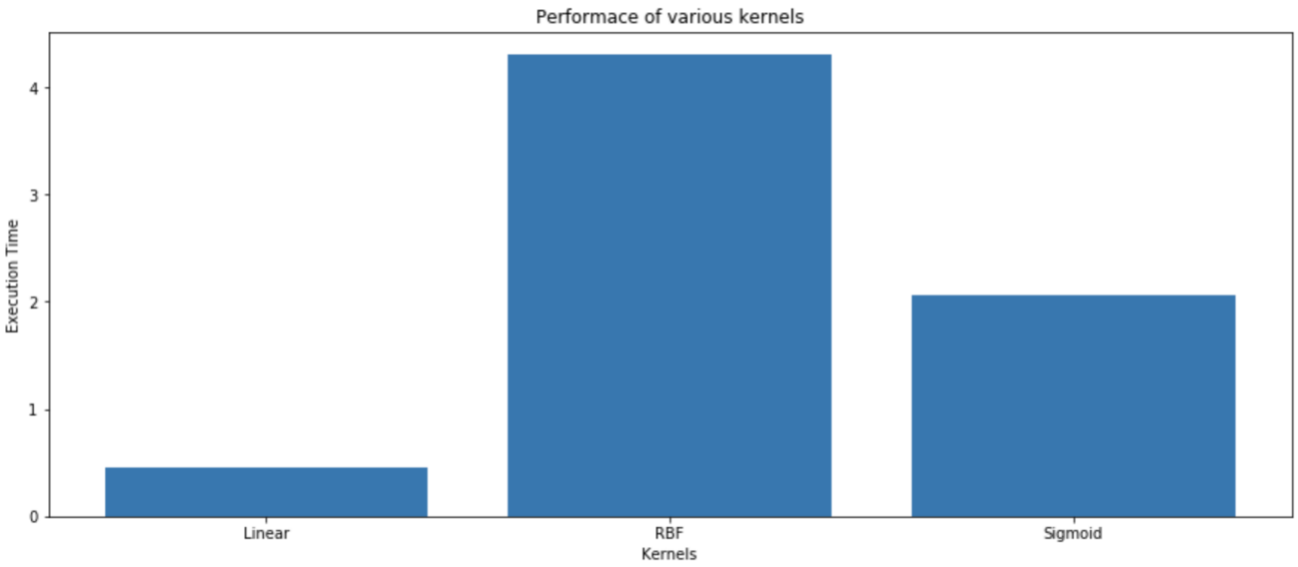
How did you pick various kernels, and how did they compare?

I tried using all kernels, but 'poly' and sometimes even 'rbf' ran for 2 hours and still did not respond
These are the performance charts of the kernel functions

Least time taken by linear svm.SVCLinear() not svm.SVC(kernel='linear')

Performance of various Kernels (Type of Kernel vs Execution time)

Churn Data set



Online News Popularity

RBF could not ran completely (I waited for 2 hours)