BUAN 6341 APPLIED MACHINE LEARNING ASSIGNMENT 2

Support Vector Machines (S.V.M.), Decision Trees & Boosting Algorithms

By Arpit Chaukiyal (axc176630)

Datasets Used:

1. Student Performance datasets (https://archive.ics.uci.edu/ml/datasets/Student+Performance#)
2. Telecom Churn Dataset (Kaggle)

Scripting Language Used: Python

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 publicly available dataset in Kaggle. It has 21 features and 7043 observations. All the data features have non-Null except "TotalCharges". TotalCharges has 11 null values, which I have deleted as it contributes less than 1% of data. It has 3 numerical variables and rest all 18 features are categorical variables.
   During the data processing phase I have converted all the categorical variables into dummy variables and normalized the categorical variables.

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.

   I have used the svm package of sklearn.

3. 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).

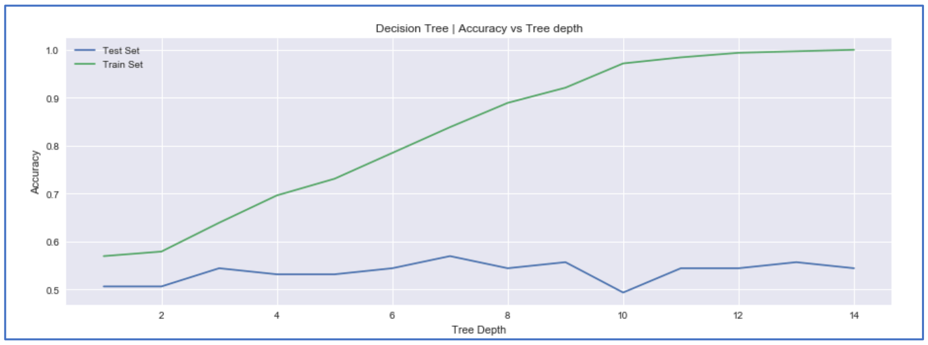
   I have used the linear kernel, RBF kernel, and Sigmoid kernel.

4. Download and use any decision trees package to classify your classification problems.

   I have used DecisionTreeClassifier form sklearn

5. 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.

   <u>Student Dataset</u>:
   Initial Model: (Using Gini Index)
   The initial accuracy is 54.43%.
   I have plotted the learning curve as accuracy with respect of the depth of the tree, usinf gini index

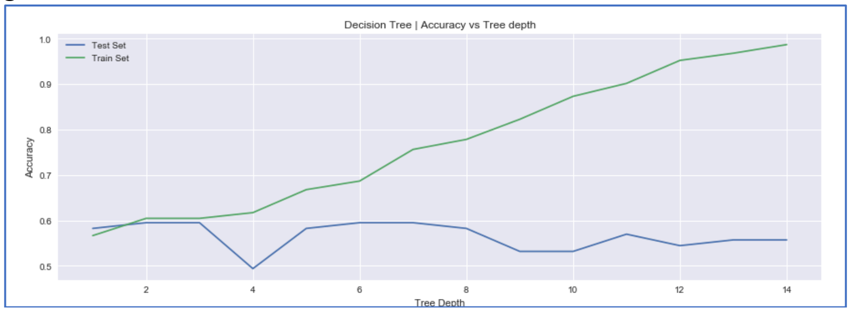


Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.

The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 2.

Model With Pruning (depth = 2)
Accuracy = 56.962%
The accuracy of this model increased although the magnitude is not so good.

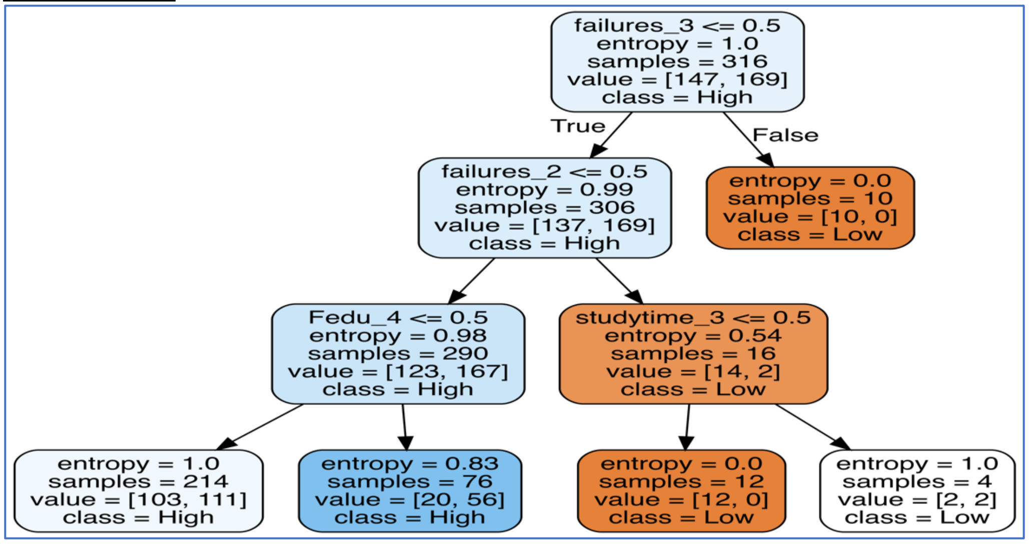
Initial Model: (Using InfoGain)
The initial accuracy is 55.69%.
I have plotted the learning curve as accuracy with respect of the depth of the tree, using info gain.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 3.

Model With Pruning (depth = 3)
Accuracy = 59.49%

Decision Tree:

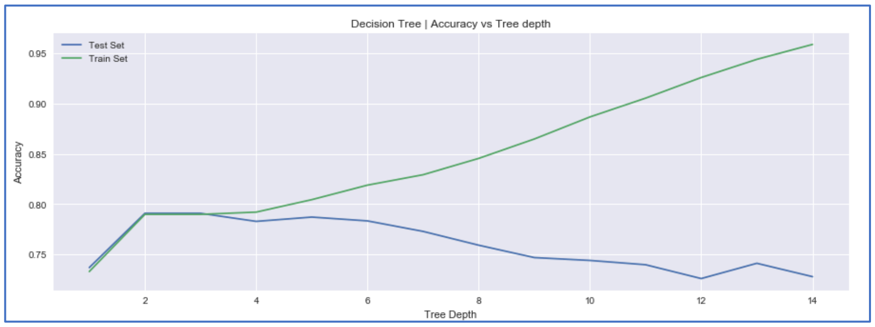


__**Churn Dataset**__:

Initial Model: (Using Gini Index)

The initial accuracy is 70.758%.

I have plotted the learning curve as accuracy with respect of the depth of the tree, usinf gini index.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.

The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.

From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 3.

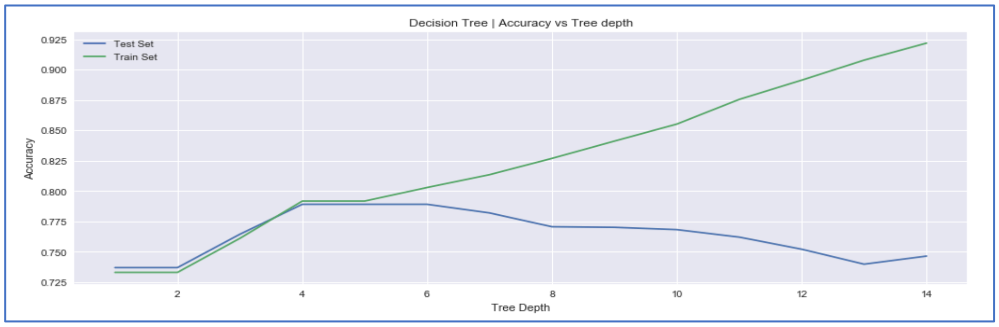
Model With Pruning (depth = 3)

Accuracy = 79.10%

Initial Model: (Using InfoGain)

The initial accuracy is 72.41%

I have plotted the learning curve as accuracy with respect of the depth of the tree, using info gain.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.

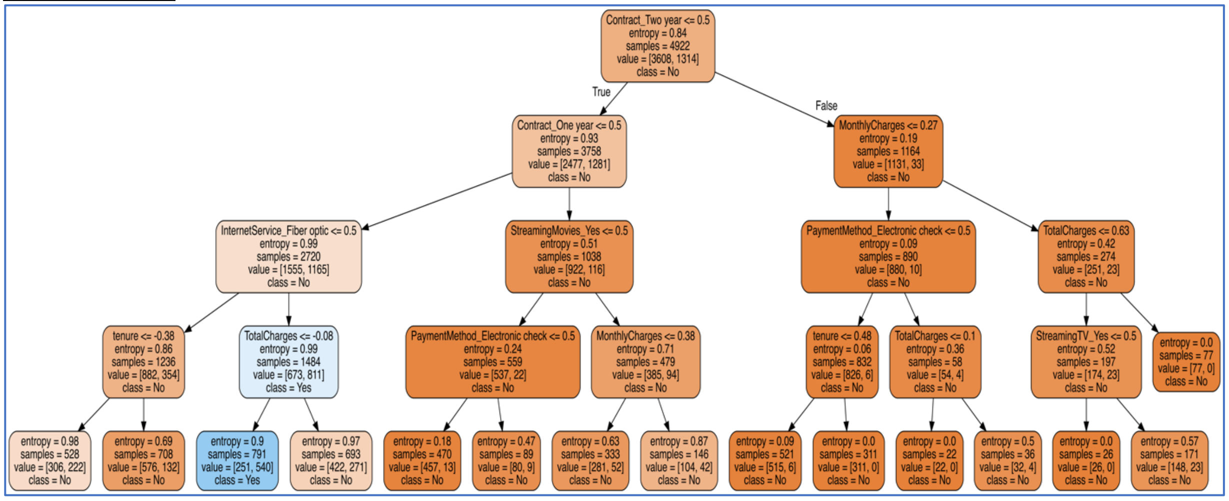
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.

From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 4.

Model With Pruning (depth = 4)

Accuracy = 78.910%

Decision Tree:


Contract_Two year <= 0.5 entropy = 0.84 samples = 4922 value = [3608, 1314] class = No
True
False
Contract_One year <= 0.5 entropy = 0.93 samples = 3758 value = [2477, 1281] class = No
MonthlyCharges <= 0.27 entropy = 0.19 samples = 1164 value = [1131, 33] class = No
InternetService_Fiber optic <= 0.5 entropy = 0.99 samples = 2720 value = [1555, 1165] class = No
StreamingMovies_Yes <= 0.5 entropy = 0.51 samples = 1038 value = [922, 116] class = No
PaymentMethod_Electronic check <= 0.5 entropy = 0.09 samples = 890 value = [880, 10] class = No
TotalCharges <= 0.63 entropy = 0.42 samples = 274 value = [251, 23] class = No
tenure <= -0.38 entropy = 0.86 samples = 1236 value = [882, 354] class = No
TotalCharges <= -0.08 entropy = 0.99 samples = 1484 value = [673, 811] class = Yes
PaymentMethod_Electronic check <= 0.5 entropy = 0.24 samples = 559 value = [537, 22] class = No
MonthlyCharges <= 0.38 entropy = 0.71 samples = 479 value = [385, 94] class = No
tenure <= 0.48 entropy = 0.06 samples = 832 value = [826, 6] class = No
TotalCharges <= 0.1 entropy = 0.36 samples = 58 value = [54, 4] class = No
StreamingTV_Yes <= 0.5 entropy = 0.52 samples = 197 value = [174, 23] class = No
entropy = 0.0 samples = 77 value = [77, 0] class = No
entropy = 0.98 samples = 528 value = [306, 222] class = No
entropy = 0.69 samples = 708 value = [576, 132] class = No
entropy = 0.9 samples = 791 value = [251, 540] class = Yes
entropy = 0.97 samples = 693 value = [422, 271] class = No
entropy = 0.18 samples = 470 value = [457, 13] class = No
entropy = 0.47 samples = 89 value = [80, 9] class = No
entropy = 0.63 samples = 333 value = [281, 52] class = No
entropy = 0.87 samples = 146 value = [104, 42] class = No
entropy = 0.09 samples = 521 value = [515, 6] class = No
entropy = 0.0 samples = 311 value = [311, 0] class = No
entropy = 0.0 samples = 22 value = [22, 0] class = No
entropy = 0.5 samples = 36 value = [32, 4] class = No
entropy = 0.0 samples = 26 value = [26, 0] class = No
entropy = 0.57 samples = 171 value = [148, 23] class = No

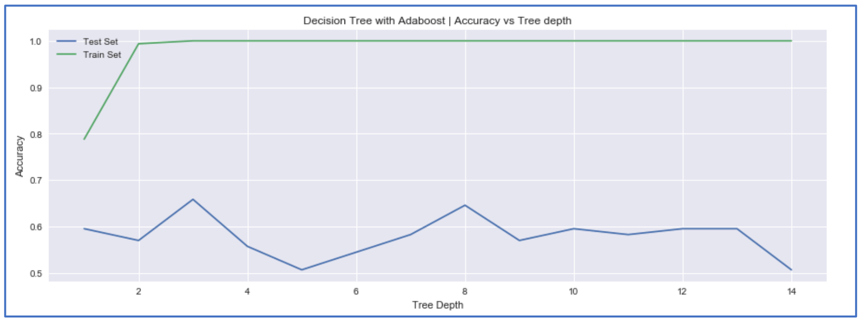
I tried changing: gini to entropy, but it doesn't appear to matter.
On further investigation, I found out:

- 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

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

Student Dataset (ADA Boosting)

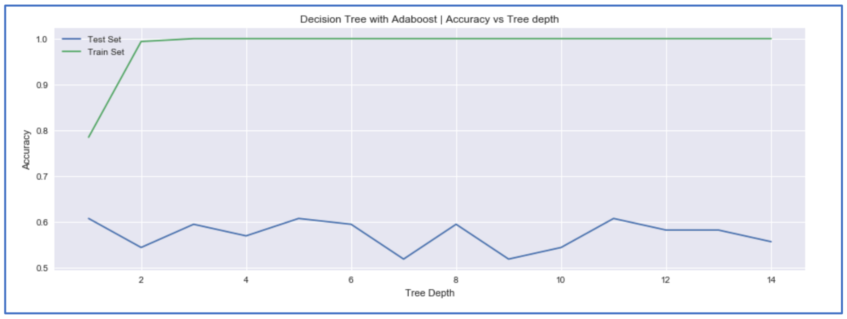
Initial Model: (Using Gini Index)
The initial accuracy is 50.63%.
I have plotted the learning curve as accuracy with respect of the depth of the tree, usinf gini index



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 3.

Model With Pruning (depth = 3)
Accuracy = 60.759%
The accuracy of this model increased although the magnitude is not so good.

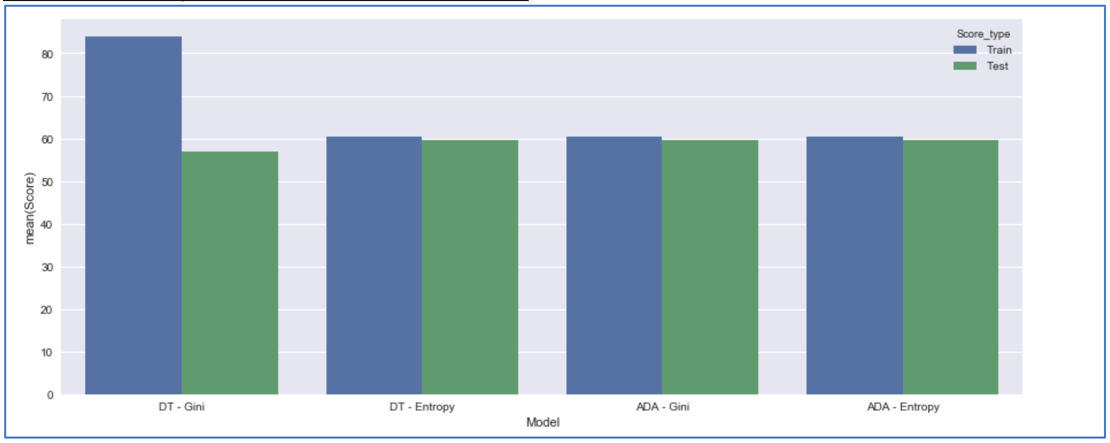
Initial Model: (Using InfoGain)
The initial accuracy is 55.696%.
I have plotted the learning curve as accuracy with respect of the depth of the tree, using info gain.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 1.

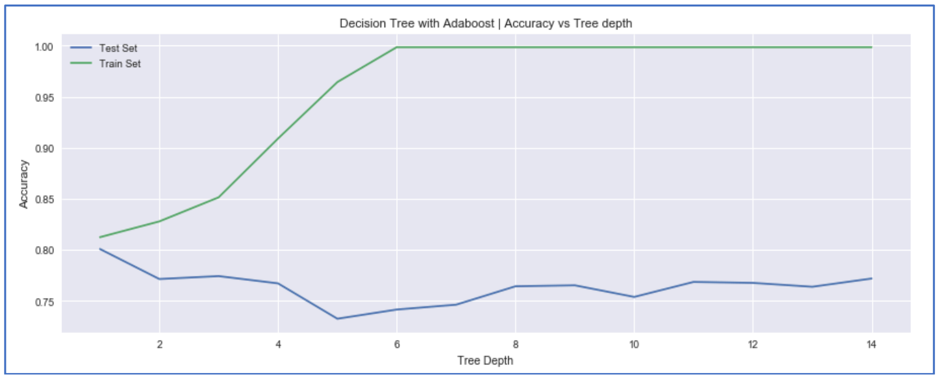
Model With Pruning (depth = 1)
Accuracy = 60.75%

<u>All Model comparisons for Student Dataset</u>:



<u>Churn Dataset</u>:

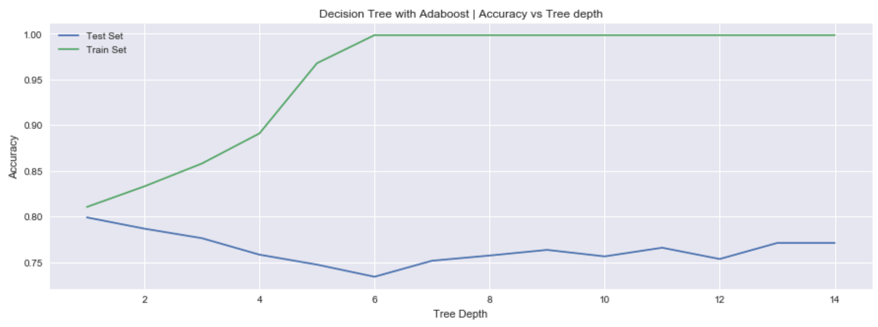
Initial Model: (Using Gini Index)
The initial accuracy is 71.943%.
I have plotted the learning curve as accuracy with respect of the depth of the tree, usinf gini index.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 1.

Model With Pruning (depth = 1)
Accuracy = 80.095%

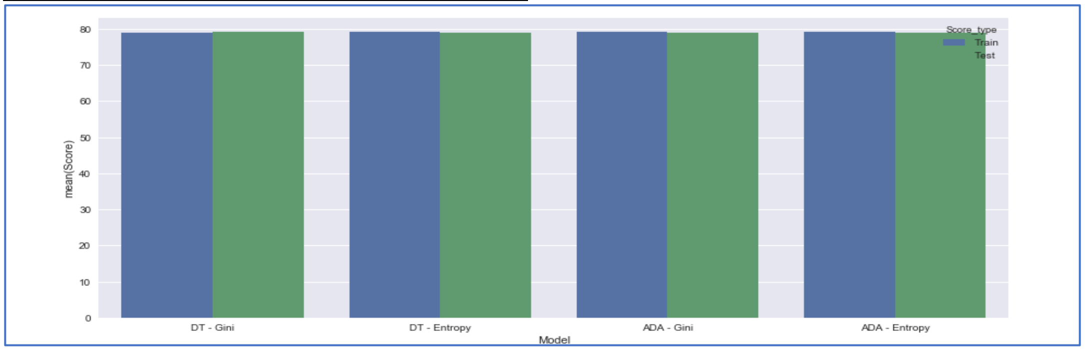
Initial Model: (Using InfoGain)
The initial accuracy is 75.877%
I have plotted the learning curve as accuracy with respect of the depth of the tree, using info gain.



Findings: The accuracy of the train dataset increases as the tree depth increases and eventually the it reaches close to 1, which results to the overfitting of the model.
The accuracy of the test dataset increases as the depth increases but after certain depth is starts to decreases, which is expected.
From the graph of accuracy as the function of tree depth, it as clear that for this dataset the optimum tree depth is 1.

Model With Pruning (depth = 1)
Accuracy = 79.905%

<u>All Model comparisons for Student Dataset</u>

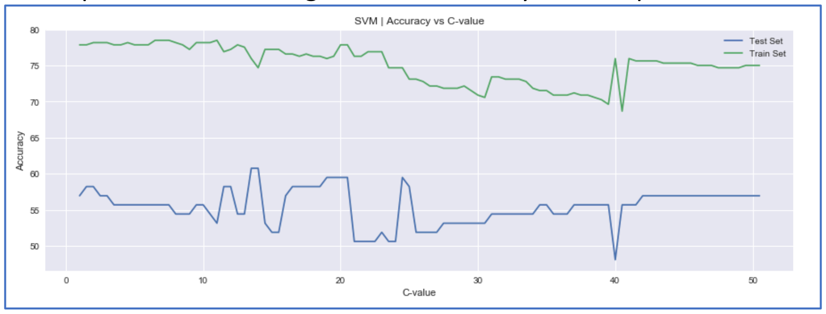


Performance of all these boosting algorithms are almost comparable.

7. Performance comparisons (learning curves, confusion matrices, etc.) of various functions/parameters for the algorithms (e.g. kernels in SVM, pruning in decision trees, etc.) on both the data sets.

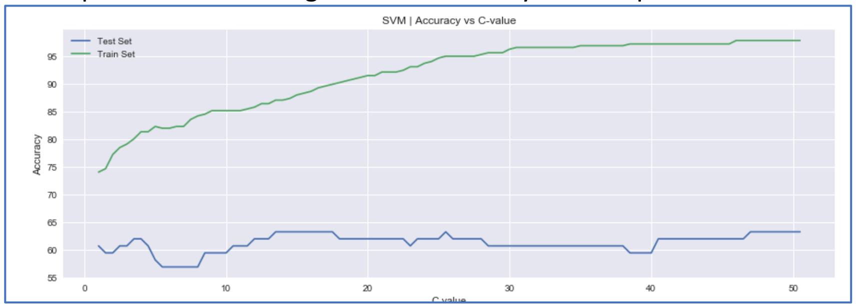
<u>Student Dataset</u>

Model: <u>(SVM Linear Kernel)</u>
The initial accuracy is 60.76%.
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 13.5.

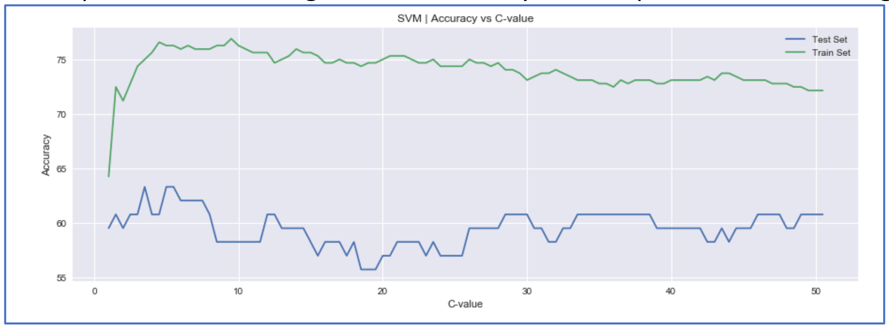
Model: <u>(SVM RBF Kernel)</u>
The initial accuracy is 63.29%.
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 13.5

Model: <u>(SVM Sigmoid Kernel)</u>
The initial accuracy is 63.29%.

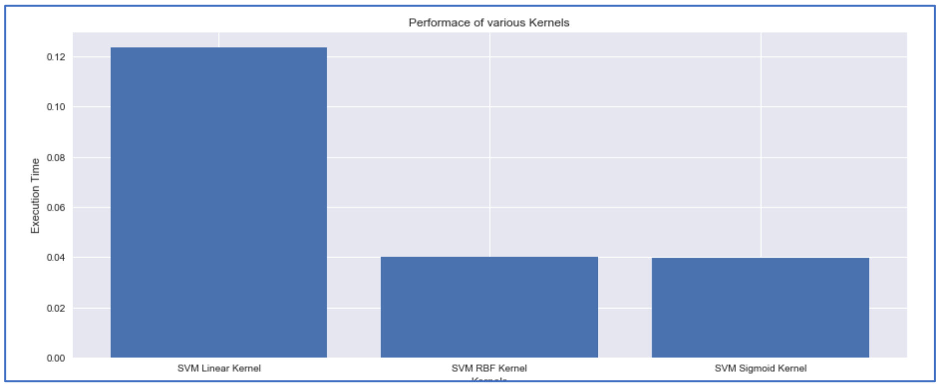
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



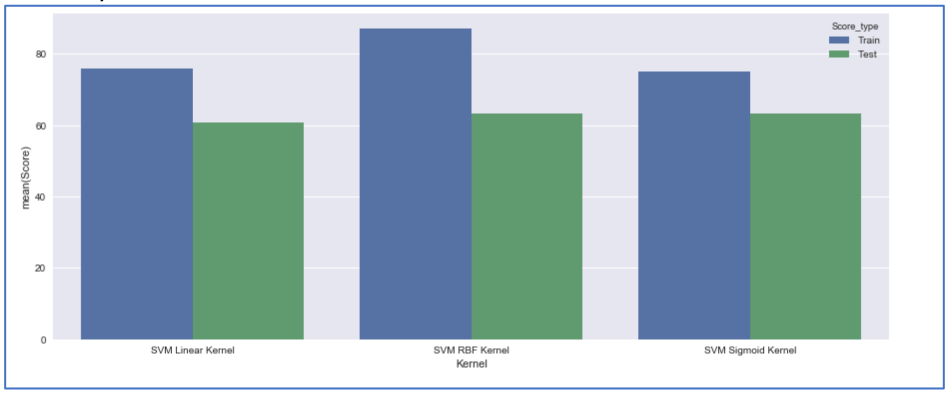
Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 13.5.

Performance of various Kernels:

- Time



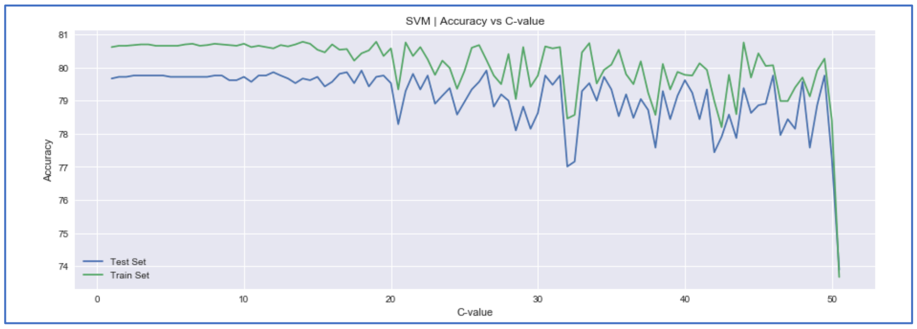
- Accuracy



Findings: As it can be seen from the above visualization that Linear Kernel is the slowest and has the poorest accuracy of all the three kernel. This may be due to the fact that our data set in not at all linearly separable. RBF kernel doest the best job of classifying the data set and has fairly less execution time.

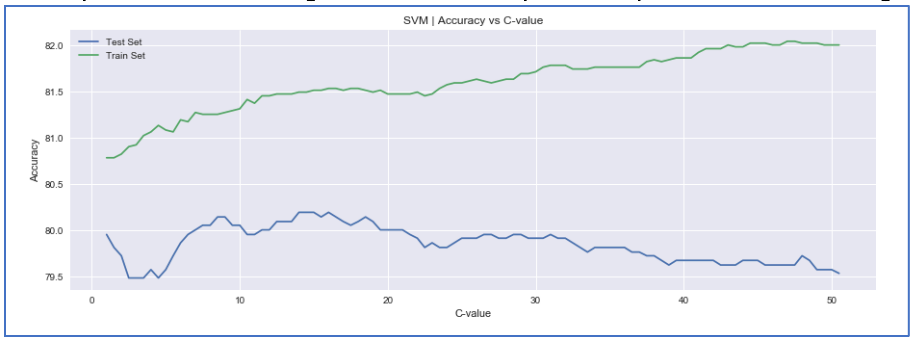
Churn Dataset:

Model: (SVM Linear Kernel)
The initial accuracy is 79.91%.
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



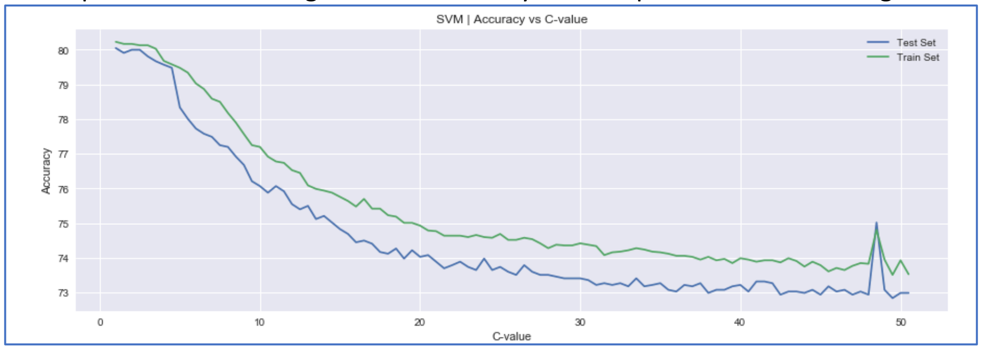
Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 18.

Model: <u>(SVM RBF Kernel)</u>
The initial accuracy is 79.95%.
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 14

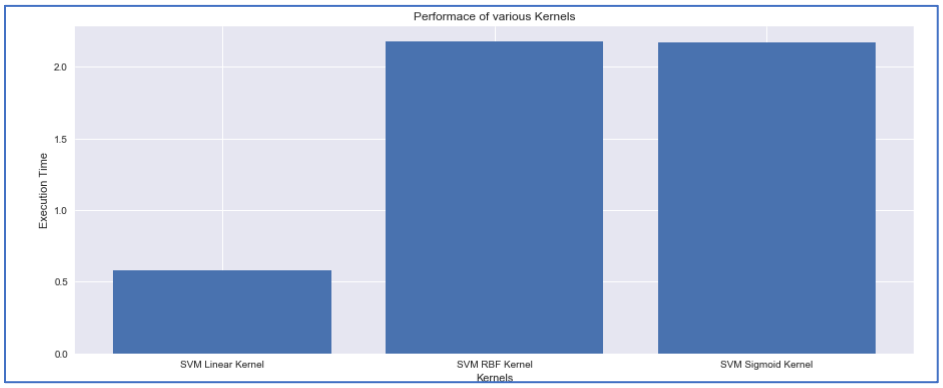
Model: <u>(SVM Sigmoid Kernel)</u>
The initial accuracy is 80.05%.
I have plotted the learning curve as accuracy with respect of the increasing value of C parameter



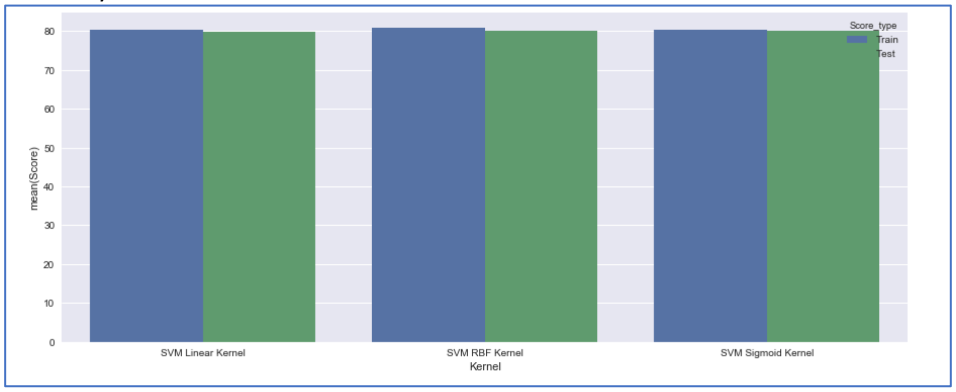
Findings: The accuracy of the train dataset decreases as the value of C parameter increases. From the learning graph above the maximum accuracy is at C = 1.

<u>Performance of various Kernels</u>:

- Time



- Accuracy



Findings: As it can be seen from the above visualization that Linear Kernel is the fastest as opposed to the student dataset and all the kernels have almost same accuracy. This may be due to the fact that our data set is best linearly separable. RBF kernel and Sigmoid function does the job with same accuracy but the execution time is very large.

8. Compare and contrast the different algorithms.

   Above are the different visualization that compare the different algorithms.

9. What additional things can you do to get better results? Would cross validation help? If yes_ then why didn't you implement it?
   Yes cross validation would help in eleiminating the sample biasing. I did not implemented it because of the time constrains.

10. How did you do pruning and when and why did you decide to stop? How did you pick various kernels, and how did they compare? Which algorithm performed the best and why? How did you define best?

    All these question have been discussed in the description of the reportabove