Prepared by: KHUSHBOO BHATIA (kxb190007) Date: 03/01/2020

BUAN 6341 Applied Machine Learning

ASSIGNMENT NO 2

Introduction:

In this report, I have implemented Support Vector Machines, Decision Trees and Boosting with Python and Scikit-Learn and build a classifier to predict whether or not it will rain tomorrow in Australia. I have used the **Rain in Australia** dataset as my second dataset for this project. This Data set is obtained from Kaggle, link to the dataset is given below –

https://www.kaggle.com/jsphyg/weather-dataset-rattle-package

This dataset contains about 10 years of daily weather observations from numerous Australian weather stations. The dataset is interesting to work on as it has many categorical features and lots of missing values to deal with.

About the Dataset 2:

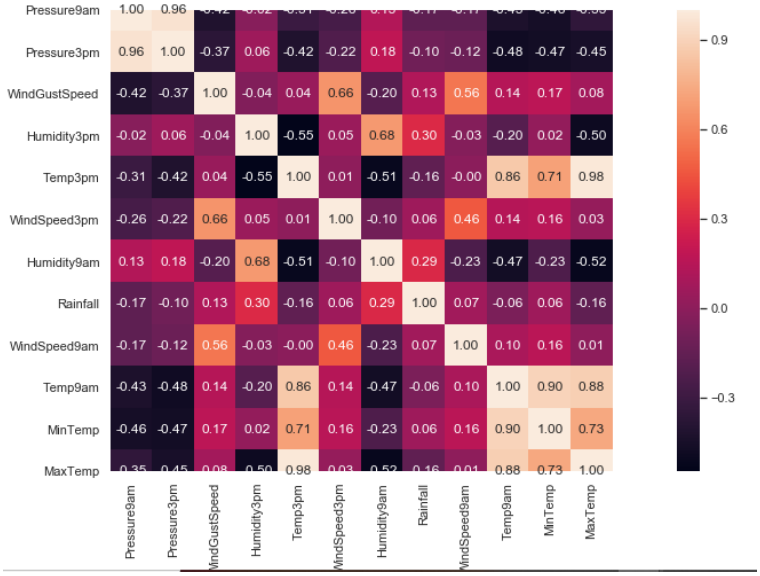
Below is the brief description about the dataset 2 i.e. Rain in Australia –

- This dataset has 142193 instances and 24 variables.
- We have excluded the variable Risk-MM while training our model.
- The predictor variable is 'RainTomorrow' which has 2 unique values 'Yes' or 'No'. Number of days no rain tomorrow is 110316 and Number of days rain tomorrow is 31877.
- Out of the total number of 'RainTomorrow' values, No appears 77.58% times and Yes appears 22.42% times.
- There are 6 categorical variables i.e. Location, WindGustDir, WindDir9am, WindDir3pm, RainToday and RainTomorrow.
- Below is the list of missing values in our dataset –

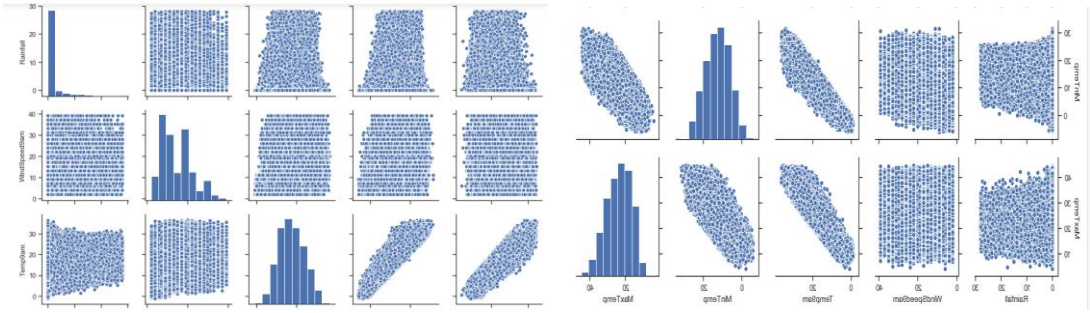
	Total	Percent
Sunshine	67816	0.476929
Evaporation	60843	0.427890
Cloud3pm	57094	0.401525
Cloud9am	53657	0.377353
Pressure9am	14014	0.098556
Pressure3pm	13981	0.098324
WindDir9am	10013	0.070418
WindGustDir	9330	0.065615
WindGustSpeed	9270	0.065193
WindDir3pm	3778	0.026570

Data Preparation and Exploratory Data Analysis:

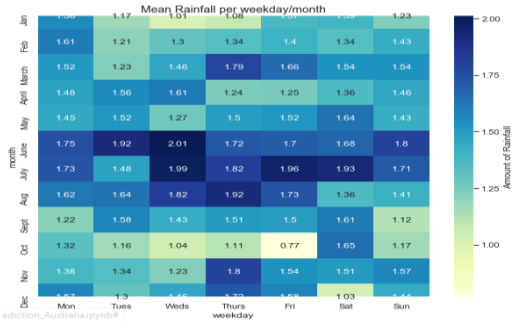
Correlation Heatmap of Rain: MinTemp and MaxTemp variables are highly positively correlated (correlation coefficient = 0.73). MinTemp and Temp9am variables are strongly positively correlated (correlation coefficient = 0.90). Pressure9am and Pressure3pm variables are strongly positively correlated (correlation coefficient = 0.96).



Feature Correlation: To understand the relationship between highly positive correlated features, I have plotted the pair plot as below –

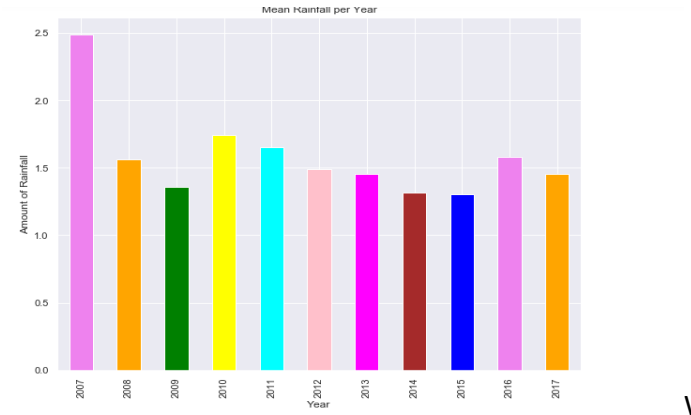


Mean Rainfall per weekday/month –



From above heatmap we can conclude that June has highest average rainfall of 2.01 mm.

Average Rainfall per year –



We can ignore year 2007 as it has only 2 months data therefore year 2010 has highest average rainfall per year.

Scaling Data –

Working with values in a wide range is not convenient, we need to scale it. In this case we are going to normalize it and scaling it in a 0-1 range.

Model and Estimation Techniques for Dataset 2:

1. Support Vector Machine (Linear Kernel)

For linear kernel the equation for prediction for a new input using the dot product between the input (x) and each support vector (xi) is calculated as follows:

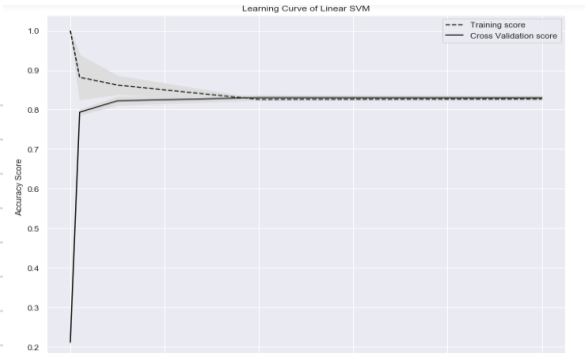
K(x, xi) = sum(x * xi)

For this experiment, I have taken below hyperparameters:

C: [0.1, 1, 10, 100] and gamma: [1, 0.1, 0.01, 0.001, 0.0001]

Using grid search and above hyperparameters values of C and gamma, SVM Linear kernel is used. There are 20 (4*5) possible combinations for above parameters.

Linear Kernal	
Best Param C	10
Best Param Gamma	1
Train Set Accuracy	85.80%
Test Set Accuracy	84.56%
Train Error	0.142
Test Error	0.1543



Conclusion: Both the Train and Test accuracies are almost same. Increasing the value of C results in higher test set accuracy and a slightly increased training set accuracy. So, we can conclude that a more complex model should perform better.

From learning curve of Linear SVM, the gap between the training and cross validation accuracy score decreases. Therefore, the model is generalizing as the training set increases.

2. Support Vector Machine (Gaussian Kernel/Radial Basis Function)

Gaussian RBF (Radial Basis Function) is another popular Kernel method used in SVM models for more. RBF kernel is a function whose value depends on the distance from the origin or from some point. Gaussian Kernel is of the following format:

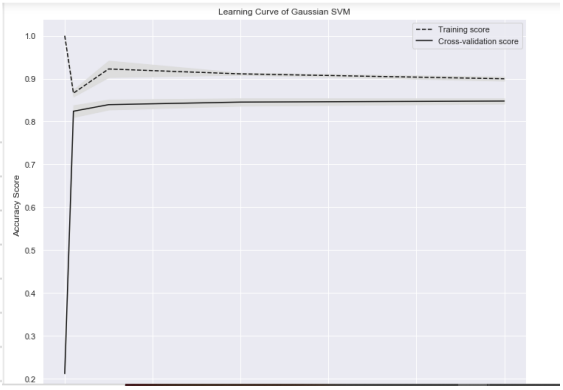
K(x, xi) = exp(-gamma * sum((x – xi^2))

For this experiment, I have taken below hyperparameters:

C: [0.1, 1, 10, 100] and gamma: [1, 0.1, 0.01, 0.001, 0.0001]

Using grid search and above hyperparameters values of C and gamma, SVM Gaussian kernel is used. There are 20 (4*5) possible combinations for above parameters.

Gaussian Kernal	
Best Param C	100
Best Param Gamma	0.01
Train Set Accuracy	87.10%
Test Set Accuracy	85.04%
Train Error	0.129
Test Error	0.1495



Conclusion: Gaussian SVM has better accuracy compared to the Linear SVM.

From learning curve of Gaussian SVM, there is narrow gap between the training and cross validation accuracy score.

3. Support Vector Machine (Polynomial Kernel)

In the polynomial kernel, we simply calculate the dot product by increasing the power of the kernel. The polynomial kernel is defined as:

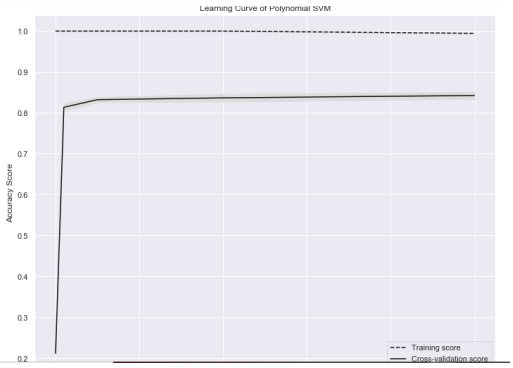
K(x, xi) = 1 + sum(x * xi)^d

For this experiment, I have taken below hyperparameters:

C= [0.1, 1, 10], gamma= [1, 0.1, 0.01, 0.001], Degree = [3, 5]

Using grid search and above hyperparameters values of C, gamma and Degree, SVM Polynomial kernel is used. There are 24 (3*2*4) possible combinations for above parameters.

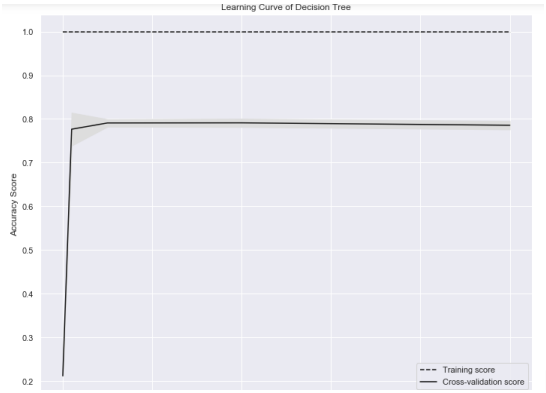
Polynomial Kernal	
Best Param C	10
Best Param Gamma	0.1
Best Param Degree	3
Train Set Accuracy	89.30%
Test Set Accuracy	84.52%
Train Error	0.1073
Test Error	0.1548



Conclusion: Polynomial SVM model is the best as compare to Linear SVM and Gaussian SVM.

From learning curve of Polynomial SVM, there is gap between the training and cross validation accuracy score. The model is not generalized and suffering from high variance.

Decision Tree: Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. The train and test accuracy are 1 and 0.784 respectively. The train error is zero and test error is 0.2159.



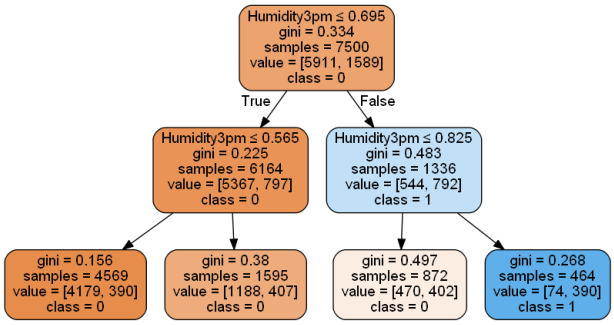
From learning curve of Decision Tree, there is huge gap between the training and cross validation accuracy score. The model is suffering from high bias.

Pruned Decision Tree: Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances. Pruning is performed using below Hyper parameters –

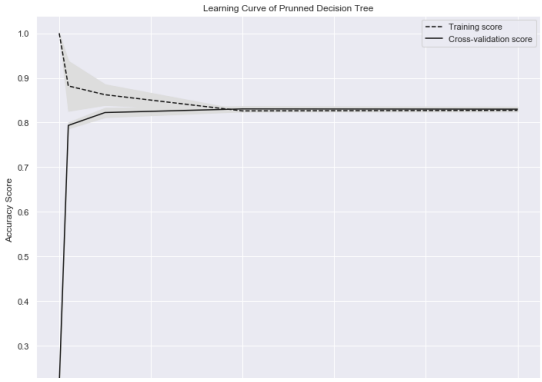
Criterion= ["gini", "entropy"], **min_samples_split** = [2, 10, 20], **max_depth** = [None, 2, 5, 10],

min_samples_leaf = [1, 5, 10], **max_leaf_nodes** = [None, 5, 10, 20]

Pruned Decision Tree	
Best Criterion	Gini
Best Max Leaf Nodes	10
Best Min Samples Leaf	1
Best Min Samples Split	20
Train Set Accuracy	83.00%
Test Set Accuracy	82.40%
Train Error	0.1073
Test Error	0.1548



Humidity3pm is the root node and has got Gini index of 0.334

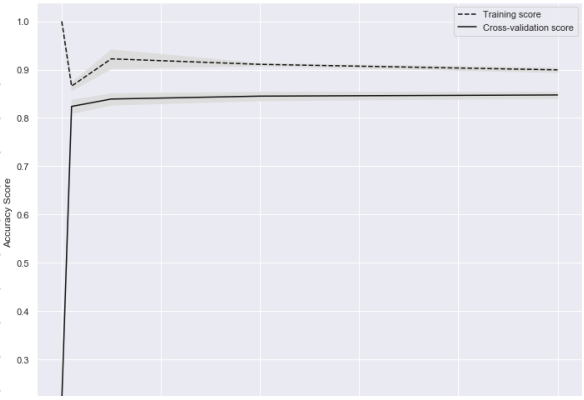


From learning curve of Pruned Decision Tree, the gap between the training and cross validation accuracy scores decreases, the model is generalizing as the training set increases.

Boosting (XG Boost): XG Boost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XG Boost is performed using below Hyper parameters –

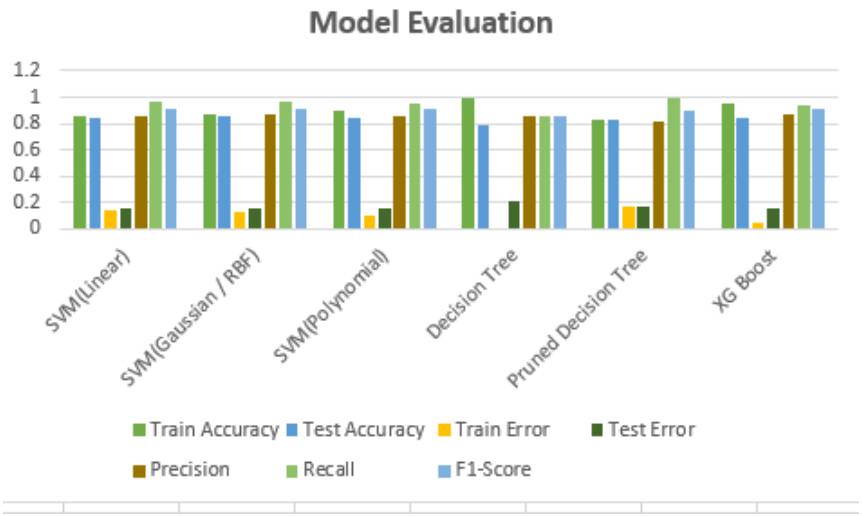
max_depth= [2,3,5,10], **n_estimators** = [100,500,1000], **learning_rate** = [0.01,0.005,0.001], **min_child_weight** = [1,5], **eta** = [.3], **gamma** = [0,1,5]

XG Boost	
Best n_estimators	500
Best min_child_weight	1
Best max_depth	10
Best learning_rate	0.01
Train Set Accuracy	95.20%
Test Set Accuracy	84.72%
Train Error	0.0475
Test Error	0.1528



Model Evaluation:

Rain in Australia							
Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error	Precision	Recall	F1-Score
SVM(Linear)	0.858	0.8456	0.142	0.1544	0.86	0.96	0.91
SVM(Gaussian / RBF)	0.871	0.8504	0.129	0.1496	0.87	0.96	0.91
SVM(Polynomial)	0.893	0.8452	0.107	0.154	0.86	0.95	0.91
Decision Tree	1	0.784	0	0.215	0.86	0.86	0.86
Pruned Decision Tree	0.83	0.824	0.169	0.176	0.82	0.99	0.9
XG Boost	0.952	0.8472	0.047	0.152	0.87	0.94	0.91



Model and Estimation Techniques for Dataset 1:

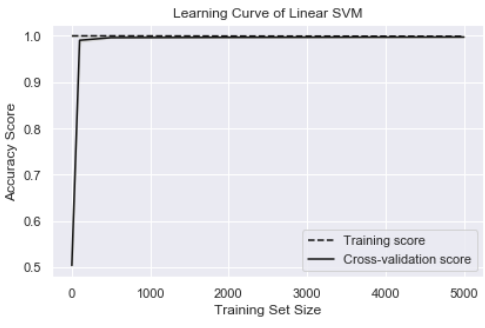
1. Support Vector Machine (Linear Kernel)

For this experiment, I have taken below hyperparameters:

C: [0.1, 1, 10, 100] and gamma: [1, 0.1, 0.01, 0.001, 0.0001]

Using grid search and above hyperparameters values of C and gamma, SVM Linear kernel is used. There are 20 (4*5) possible combinations for above parameters.

Linear Kernal	
Best Param C	100
Best Param Gamma	1
Train Set Accuracy	98.70%
Test Set Accuracy	98.61%
Train Error	0.0128
Test Error	0.0138



Conclusion: Both the Train and Test accuracies are almost same. Increasing the value of C results in higher test set accuracy and a slightly increased training set accuracy. So, we can conclude that a more complex model should perform better.

From learning curve of Linear SVM, the gap between the training and cross validation accuracy score decreases. Therefore, the model is generalizing as the training set increases.

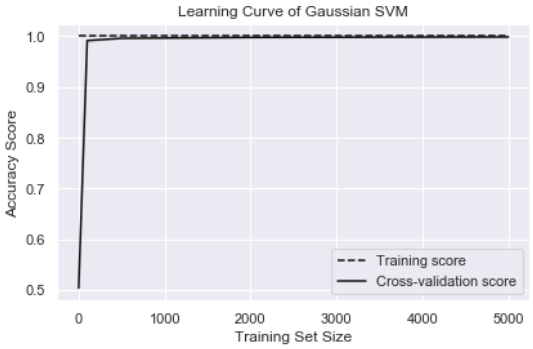
2. Support Vector Machine (Gaussian Kernel/Radial Basis Function)

For this experiment, I have taken below hyperparameters:

C: [0.1, 1, 10, 100] and gamma: [1, 0.1, 0.01, 0.001, 0.0001]

Using grid search and above hyperparameters values of C and gamma, SVM Gaussian kernel is used. There are 20 (4*5) possible combinations for above parameters.

Gaussian Kernal	
Best Param C	100
Best Param Gamma	0.1
Train Set Accuracy	98.30%
Test Set Accuracy	97.31%
Train Error	0.0166
Test Error	0.0268



From learning curve of Gaussian SVM, there is narrow gap between the training and cross validation accuracy score.

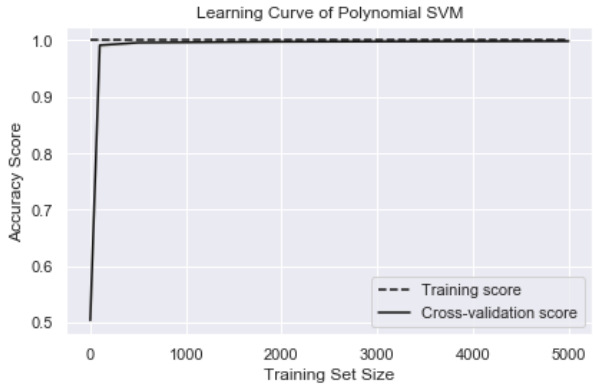
3. Support Vector Machine (Polynomial Kernel)

For this experiment, I have taken below hyperparameters:

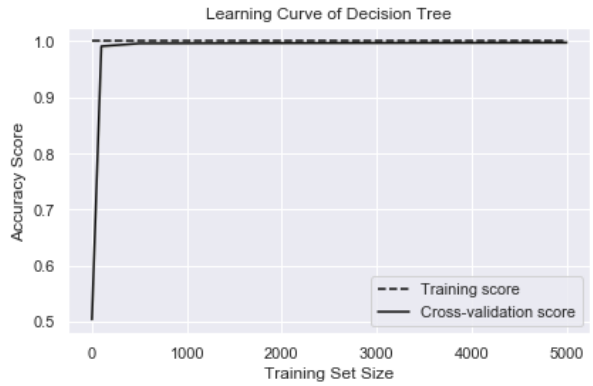
C= [0.1, 1, 10], gamma= [1, 0.1, 0.01, 0.001], Degree = [3, 5]

Using grid search and above hyperparameters values of C, gamma and Degree, SVM Polynomial kernel is used. There are 24 (3*2*4) possible combinations for above parameters.

Polynomial Kernal	
Best Param C	10
Best Param Gamma	1
Best Param Degree	3
Train Set Accuracy	99.20%
Test Set Accuracy	97.50%
Train Error	0.0082
Test Error	0.0246



Decision Tree: Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. The train and test accuracy are 1 and 1 respectively. The train and test error is zero.

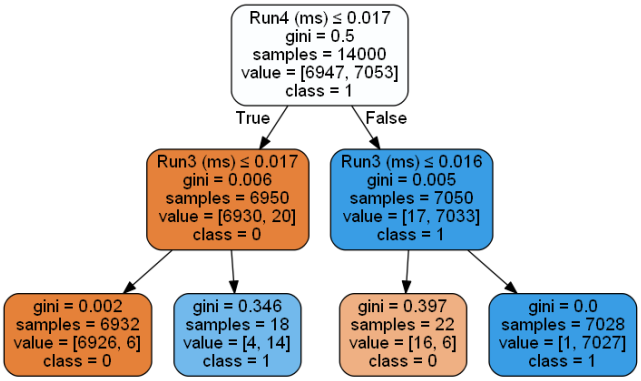


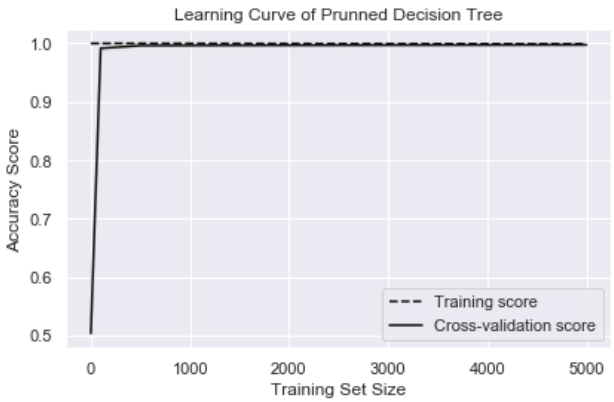
Pruned Decision Tree: Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances. Pruning is performed using below Hyper parameters –

Criterion= ["gini", "entropy"], **min_samples_split** = [2, 10, 20], **max_depth** = [None, 2, 5, 10],

min_samples_leaf = [1, 5, 10], **max_leaf_nodes** = [None, 5, 10, 20]

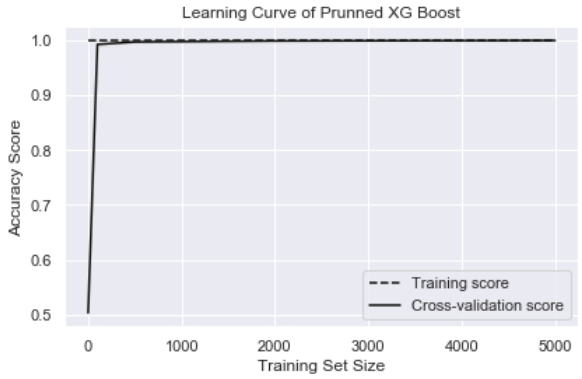
Pruned Decision Tree	
Best Criterion	Gini
Best Max Leaf Nodes	10
Best Min Samples Leaf	1
Train Set Accuracy	99.90%
Test Set Accuracy	99.88%
Train Error	0.0012
Test Error	0.0011





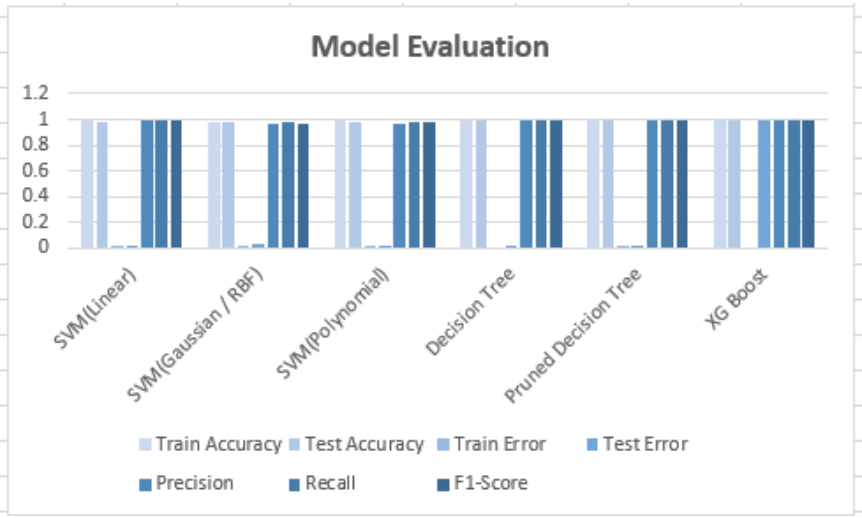
Boosting (XG Boost): XG Boost is a decision-tree-based ensemble Machine Learning algorithm that uses a gradient boosting framework. XG Boost is performed using below Hyper parameters – **max_depth**= [2,3,5,10], **n_estimators** = [100,500,1000], **learning_rate** = [0.01,0.005,0.001], **min_child_weight** = [1,5], **eta** = [.3], **gamma** = [0,1,5]

XG Boost	
Best n_estimators	1000
Best min_child_weight	1
Best max_depth	2
Best learning_rate	0.01
Train Set Accuracy	1
Test Set Accuracy	0.9995



Model Evaluation:

SGEMM GPU Kernel Performance Prediction							
Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error	Precision	Recall	F1-Score
SVM(Linear)	0.987	0.986	0.0128	0.0138	0.99	0.99	0.99
SVM(Gaussian / RBF)	0.983	0.973	0.0166	0.0268	0.97	0.98	0.97
SVM(Polynomial)	0.992	0.975	0.0082	0.0246	0.97	0.98	0.98
Decision Tree	1	0.999	0	0.0003	1	1	1
Pruned Decision Tree	0.999	0.998	0.0012	0.0011	1	1	1
XG Boost	1	0.9995	0	0.9995	1	1	1



Conclusion:

- Cross validation significantly reduces bias as most of the data is being used for fitting and it also reduces variances as most of the data also used in validation set.
- Support Vector Machine (Gaussian Kernel) performed best in terms of Accuracy, Precision and Recall for both the data sets.
- Important assumption here is that I have used subset of data out of full dataset due to system constraint.