Assignment 2: SVM, Decision tree, Boosting, pruning, cross validation

Dataset – Average GPU runtime:

The dataset (SGEMM GPU kernel performance) dataset can be downloaded at:
https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance#
There are 14 parameters. The first 4 are ordinal and the last four variables are binary. The dataset has total 241600 data entries and 18 features with the last four being the runtime measurement.

Converted the dataset into categorical dataset by categorizing the data into high and low processing. Used the value of average_gpu_runtime > 70ms as 70ms lies in the 50th percentile range it is at the center of the major amount of readings. So this would be a good value to categorize the dataset.

Divided the dataset into 60-40 ratio dataset.

SVM Model:

Linear Kernel:
Parameters tested:
C = 0.1, 1, 10
gamma = 1, 0.1, 0.01
kernel = 'Linear'
Best result: C = 1, gamma = 1, kernel = 'Linear'
Best train accuracy: 0.827
Best test accuracy: 0.8306

```
[[10515  1624]
 [ 2473  9577]]
Accuracy: 0.8306254909256273
              precision    recall  f1-score   support

           0       0.81      0.87      0.84     12139
           1       0.86      0.79      0.82     12050

    accuracy                           0.83     24189
   macro avg       0.83      0.83      0.83     24189
weighted avg       0.83      0.83      0.83     24189
```

Gaussian Kernel:
Parameters tested:
C = 0.1, 1, 10
gamma = 1, 0.1, 0.01
kernel = 'rbf'
Best result: C = 10, gamma = 0.1, kernel = 'rbf'.
Best train accuracy: 0.922
Best test accuracy: 0.9022

```
[[11303   836]
 [ 1528 10522]]
Accuracy: 0.9022696266898177
              precision    recall  f1-score   support

           0       0.88      0.93      0.91     12139
           1       0.93      0.87      0.90     12050

    accuracy                           0.90     24189
   macro avg       0.90      0.90      0.90     24189
weighted avg       0.90      0.90      0.90     24189
```

Polynomial kernel:
Parameters tested:
C = 0.1, 1, 10;
gamma = 1, 0.1, 0.01;
degree = 2, 3;
kernel = 'poly'
Best result: C = 10, degree= 3, gamma = 0.1, kernel = poly
Best train accuracy: 0.884
Best test accuracy: 0.8801

```
[[11133  1006]
 [ 1893 10157]]
Accuracy: 0.8801521352680971
              precision    recall  f1-score   support

           0       0.85      0.92      0.88     12139
           1       0.91      0.84      0.88     12050

    accuracy                           0.88     24189
   macro avg       0.88      0.88      0.88     24189
weighted avg       0.88      0.88      0.88     24189
```

Decision tree:
Performed **5 fold cross-validation** for all descision trees.

Best train accuracy: 0.973
Best test accuracy: 0.8813

```
[[10824  1315]
 [ 1554 10496]]
Accuracy: 0.8813923684319319
              precision    recall  f1-score   support

           0       0.87      0.89      0.88     12139
           1       0.89      0.87      0.88     12050

    accuracy                           0.88     24189
   macro avg       0.88      0.88      0.88     24189
weighted avg       0.88      0.88      0.88     24189
```

Pruned Decision tree:
Tested the data set with two methods of partition (gini, entropy). Entropy gave the better result. Entropy works better for the dataset if it is an imbalanced dataset and since we the dataset has a good balance of both 1s and 0s class, gini index gave the better result.

Parameters tested:
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
Best result: Criterion = gini, min_samples_split = 10, max_depth= 10, max_leaf_nodes = None, min_sample_split = 20
Best train accuracy: 0.695
Best test accuracy: 0.6974

```
[[9522 2617]
 [4702 7348]]
Accuracy: 0.6974244491297698
              precision    recall  f1-score   support

           0       0.67      0.78      0.72     12139
           1       0.74      0.61      0.67     12050

    accuracy                           0.70     24189
   macro avg       0.70      0.70      0.69     24189
weighted avg       0.70      0.70      0.70     24189
```

XG Boost:
Objective = binary: logistic
Nthread = 4
Seed = 42
Best train accuracy: 0.930
Best test accuracy: 0.9041

```
[[11219   920]
 [ 1399 10651]]
Accuracy: 0.9041299764355699
              precision    recall  f1-score   support

           0       0.89      0.92      0.91     12139
           1       0.92      0.88      0.90     12050

    accuracy                           0.90     24189
   macro avg       0.90      0.90      0.90     24189
weighted avg       0.90      0.90      0.90     24189
```

Pruned XG Boost:

Parameters tested:

N_estimators = 100, 500, 1000

Max_depth = 2, 3, 5, 10

Learning_rate = 0.01, 0.005, 0.001

Min_child_weight = 1, 5

Eta = .3

Gamma = 0, 1, 5

Best result: n_estimators = 500, min_child_weight = 5, max_depth = 10, learning_rate = 0.001, gamma = 1, eta = 0.3

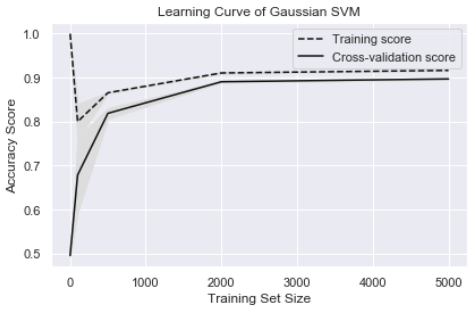
Best train accuracy: 0.978

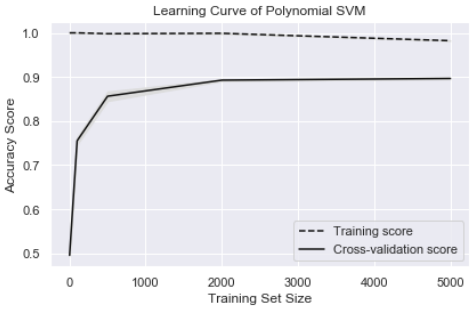
Best test accuracy: 0.9044

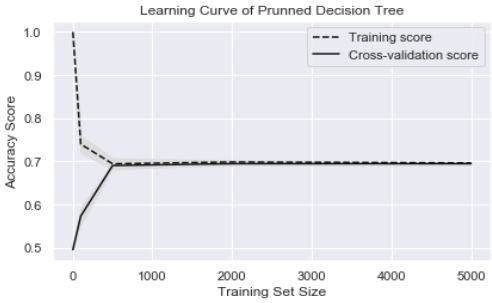
```
[[11171   968]
 [ 1344 10706]]
Accuracy: 0.904419364173798
              precision    recall  f1-score   support

           0       0.89      0.92      0.91     12139
           1       0.92      0.89      0.90     12050

    accuracy                           0.90     24189
   macro avg       0.90      0.90      0.90     24189
weighted avg       0.90      0.90      0.90     24189
```

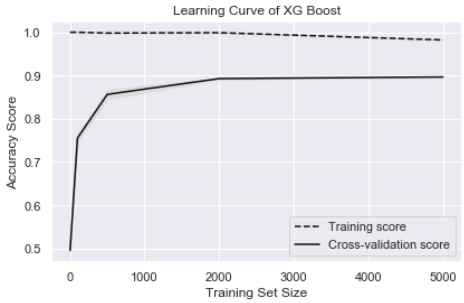
Model Performance:

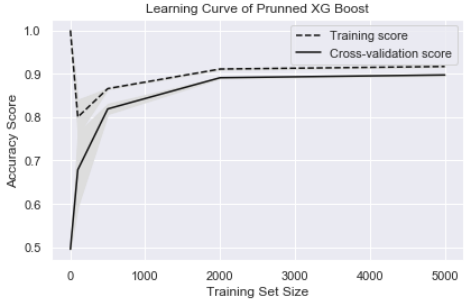
Learning curve - Training Set Size vs Accuracy





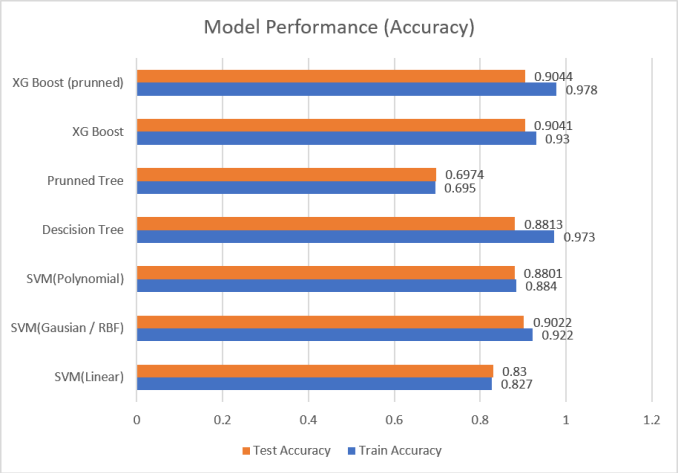


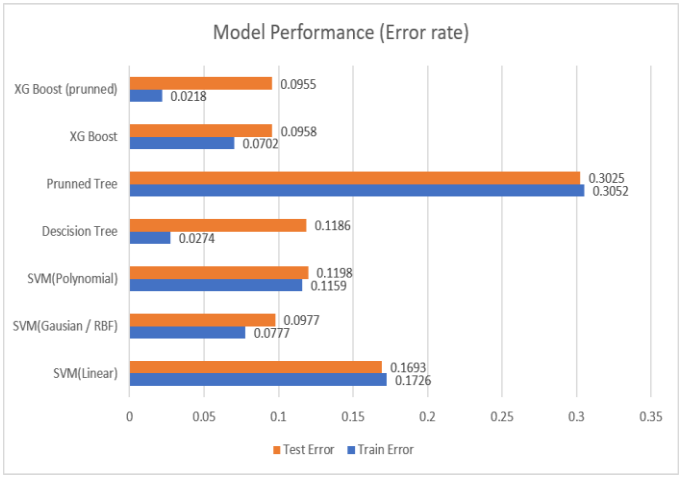




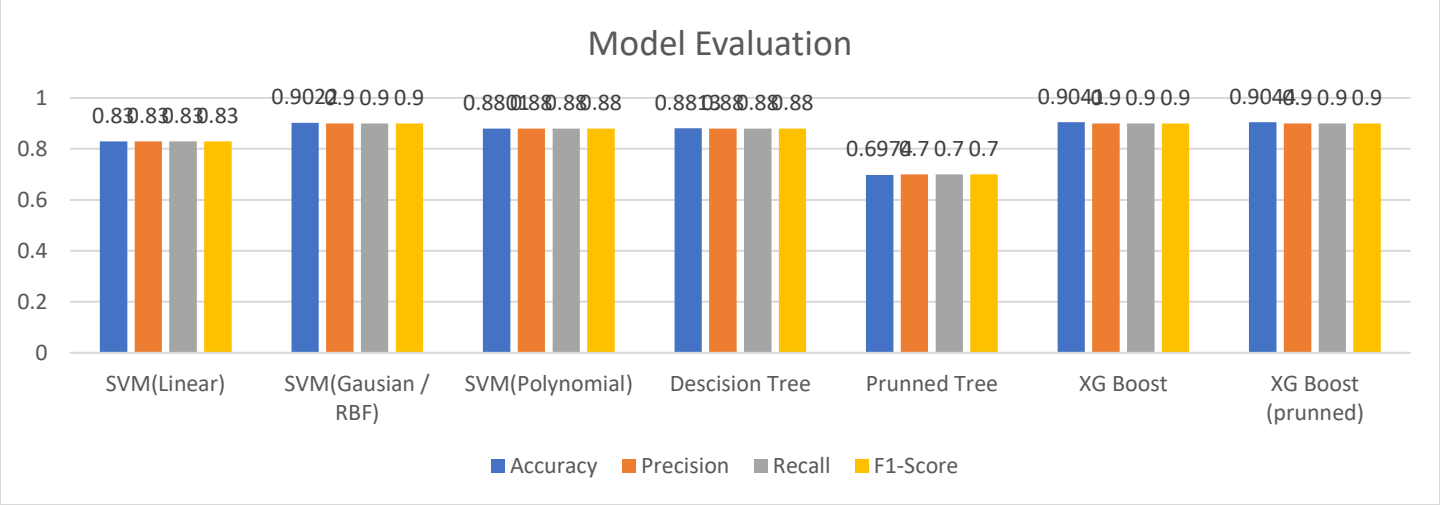
Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error
SVM(Linear)	0.827	0.83	0.1726	0.1693
SVM(Gaussian / RBF)	0.922	0.9022	0.0777	0.0977
SVM(Polynomial)	0.884	0.8801	0.1159	0.1198
Decision Tree	0.973	0.8813	0.0274	0.1186
Pruned Tree	0.695	0.6974	0.3052	0.3025
XG Boost	0.93	0.9041	0.0702	0.0958
XG Boost (pruned)	0.978	0.9044	0.0218	0.0955

- The best model among svm kernels is SVM (Gaussian / RBF) for this dataset. It gives a train and test accuracy of 0.922 and 0.9022.





Algorithm	Accuracy	Precision	Recall	F1-Score
SVM(Linear)	0.83	0.83	0.83	0.83
SVM(Gaussian / RBF)	0.9022	0.9	0.9	0.9
SVM(Polynomial)	0.8801	0.88	0.88	0.88
Decision Tree	0.8813	0.88	0.88	0.88
Pruned Tree	0.6974	0.7	0.7	0.7
XG Boost	0.9041	0.9	0.9	0.9
XG Boost (pruned)	0.9044	0.9	0.9	0.9



- XG Boost (pruned) has given the best accuracy, precision, recall and f1-score. XG Boost and SVM (Gaussian/RBF) are almost close to XG Boost (pruned).
- I haven't used the full dataset (100,000+) instead I used a subset of 60,000 out of the full dataset. I would get better result if all the instances are included in model.
- **Cross validation** reduces bias as we are using most of the data for fitting and significantly reduces variance of most of the data.
- XG Boost gives a high bias in train and test accuracy even after cross validation.

Dataset – Rains in Australia:

The dataset is obtained from Kaggle. The following is the link to the dataset. https://www.kaggle.com/jsphyg/weather-dataset-rattle-package. The weather dataset contains 142,193 daily weather observations from 49 weather stations across Australia over the period November 2007 to June 2017 and with 24 features such as Rain tomorrow, min Temp, max Temp etc. This dataset excites me because it has lot of missing values and many categorical features.

Number of days of rain tomorrow: 31877
Number of days no rain tomorrow: 110316

The dataset's final output variable is already a categorical variable with values 'yes' and 'no' for rains_tomorrow.

Divided the dataset into 70-30 ratio dataset.

SVM Model:

Linear Kernel:
Parameters tested :
C = 0.1, 1, 10, 100;
gamma = 1, 0.1, 0.01, 0.001, 0.0001;
kernel = 'Linear'
Best result: C = 1, gamma = 1, kernel = 'Linear'
Best train accuracy: 0.850
Best test accuracy: 0.846

```
[[1880   65]
 [ 320  235]]
Accuracy: 0.846
              precision    recall  f1-score   support

         0.0       0.85      0.97      0.91      1945
         1.0       0.78      0.42      0.55       555

    accuracy                           0.85      2500
   macro avg       0.82      0.70      0.73      2500
weighted avg       0.84      0.85      0.83      2500
```

Gaussian Kernel:
Parameters tested :
C = 0.1, 1, 10, 100;
gamma = 1, 0.1, 0.01, 0.001, 0.0001;
kernel = 'rbf'
Best result: C = 100, gamma = 0.01 , kernel = 'rbf'.
Best train accuracy: 0.867
Best test accuracy: 0.8456

```
[[1876   69]
 [ 317  238]]
Accuracy: 0.8456
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1945
         1.0       0.78      0.43      0.55       555

    accuracy                           0.85      2500
   macro avg       0.82      0.70      0.73      2500
weighted avg       0.84      0.85      0.83      2500
```

Polynomial kernel:
Parameters tested:
C = 0.1, 1, 10;
gamma = 1, 0.1, 0.01, 0.001;
degree = 3, 5;
kernel = 'poly'
Best result: C = 10, degree= 3, gamma = 0.1, kernel = poly
Best train accuracy: 0.888
Best test accuracy: 0.8444

```
[[1869   76]
 [ 313  242]]
Accuracy: 0.8444
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1945
         1.0       0.76      0.44      0.55       555

    accuracy                           0.84      2500
   macro avg       0.81      0.70      0.73      2500
weighted avg       0.84      0.84      0.83      2500
```

Decisoin tree:
Performed **5 fold cross-validation** for all descision trees.

Descision tree breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementely developed. The train and test accuracy are **1** and **0.7852** respectively. There is a huge gap between training and test accuracy. The model is suffering from high bias. The training error is zero and the tree has used all the features.
Best train accuracy: 1
Best test accuracy: 0.7852

```
[[1679  266]
 [ 271  284]]
Accuracy: 0.7852
              precision    recall  f1-score   support

         0.0       0.86      0.86      0.86      1945
         1.0       0.52      0.51      0.51       555

    accuracy                           0.79      2500
   macro avg       0.69      0.69      0.69      2500
weighted avg       0.78      0.79      0.78      2500
```

Pruned Decision tree:

Tested the data set with two methods of partition (gini, entropy). Entropy gave the better result as it works better for the dataset if it is an imbalanced dataset.

Parameters tested:

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

Best result: Criterion = entropy, min_samples_split = 10, max_depth= 5, max_leaf_nodes = None, min_sample_split = 2

Best train accuracy: 0.835

Best test accuracy: 0.826

```
[[1916   29]
 [ 406  149]]
Accuracy: 0.826
              precision    recall  f1-score   support

         0.0       0.83      0.99      0.90      1945
         1.0       0.84      0.27      0.41       555

    accuracy                           0.83      2500
   macro avg       0.83      0.63      0.65      2500
weighted avg       0.83      0.83      0.79      2500
```

XG Boost:

Objective = binary:logistic
Nthread = 4
Seed = 42

Best train accuracy: 0.867

Best test accuracy: 0.848

```
[[1858   87]
 [ 293  262]]
Accuracy: 0.848
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1945
         1.0       0.75      0.47      0.58       555

    accuracy                           0.85      2500
   macro avg       0.81      0.71      0.74      2500
weighted avg       0.84      0.85      0.83      2500
```

Pruned XG Boost:

Parameters tested:

N_estimators = 100, 500, 1000
Max_depth = 2, 3, 5, 10
Learning_rate = 0.01, 0.005, 0.001
Min_child_weight = 1, 5
Eta = .3
Gamma = 0, 1, 5

Best result: n_estimators = 1000, min_child_weight = 1, max_depth = 5, learning_rate = 0.01, gamma = 5, eta = 0.3

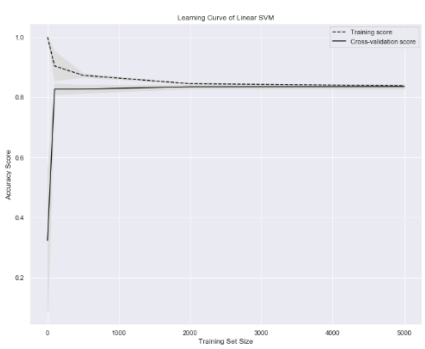
Best train accuracy: 0.913

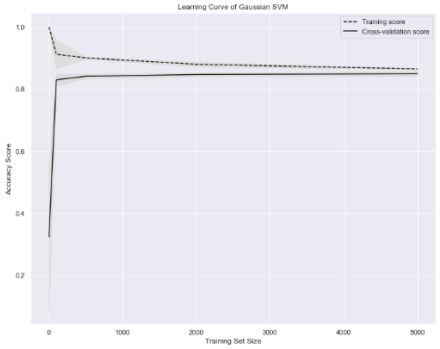
Best test accuracy: 0.85

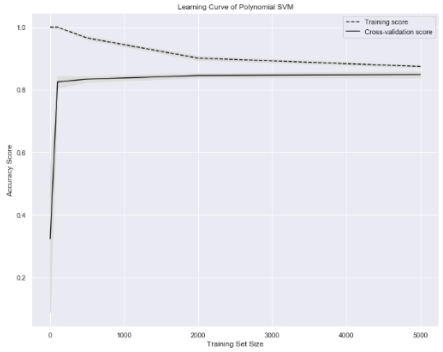
```
[[1866   79]
 [ 296  259]]
Accuracy: 0.85
              precision    recall  f1-score   support

         0.0       0.86      0.96      0.91      1945
         1.0       0.77      0.47      0.58       555

    accuracy                           0.85      2500
   macro avg       0.81      0.71      0.74      2500
weighted avg       0.84      0.85      0.84      2500
```

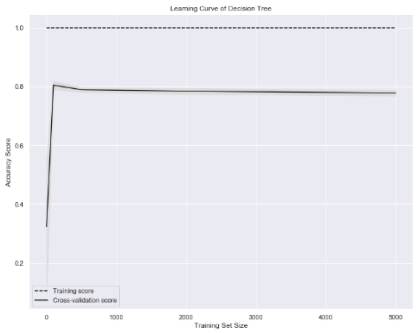
Model Performance:

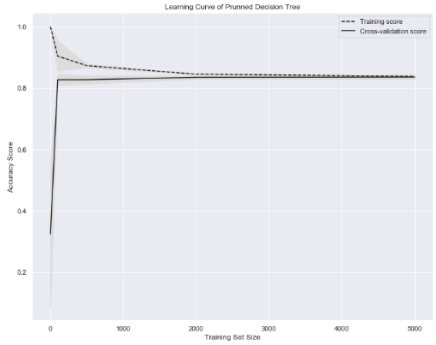
Learning curve - Training Set Size vs Accuracy

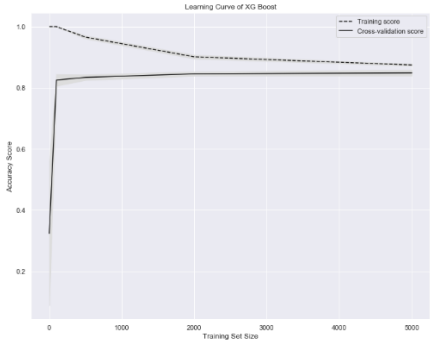


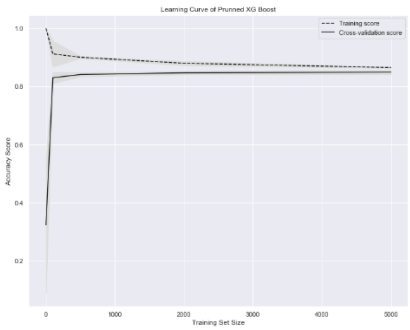






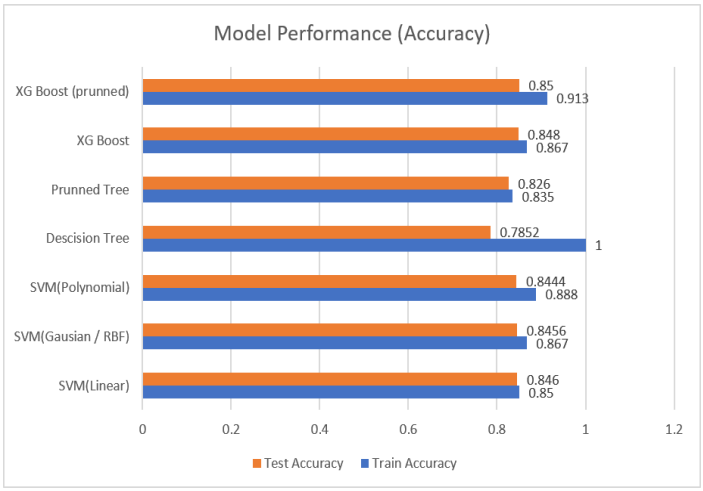


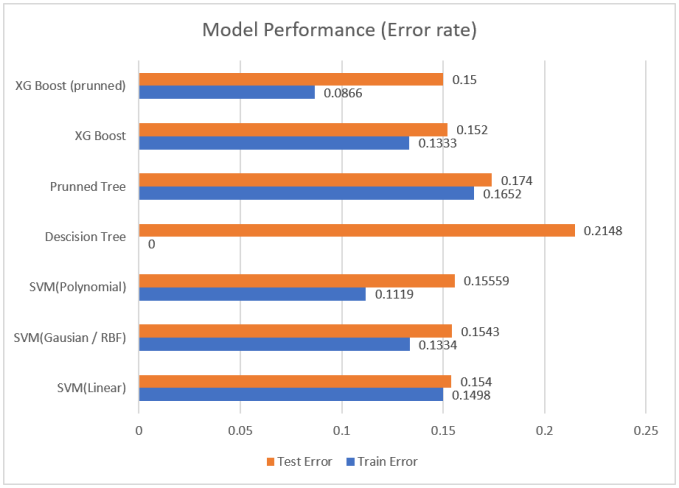




Algorithm	Train Accuracy	Test Accuracy	Train Error	Test Error
SVM(Linear)	0.85	0.846	0.1498	0.154
SVM(Gaussian / RBF)	0.867	0.8456	0.1334	0.1543
SVM(Polynomial)	0.888	0.8444	0.1119	0.15559
Decision Tree	1	0.7852	0	0.2148
Pruned Tree	0.835	0.826	0.1652	0.174
XG Boost	0.867	0.848	0.1333	0.152
XG Boost (pruned)	0.913	0.85	0.0866	0.15

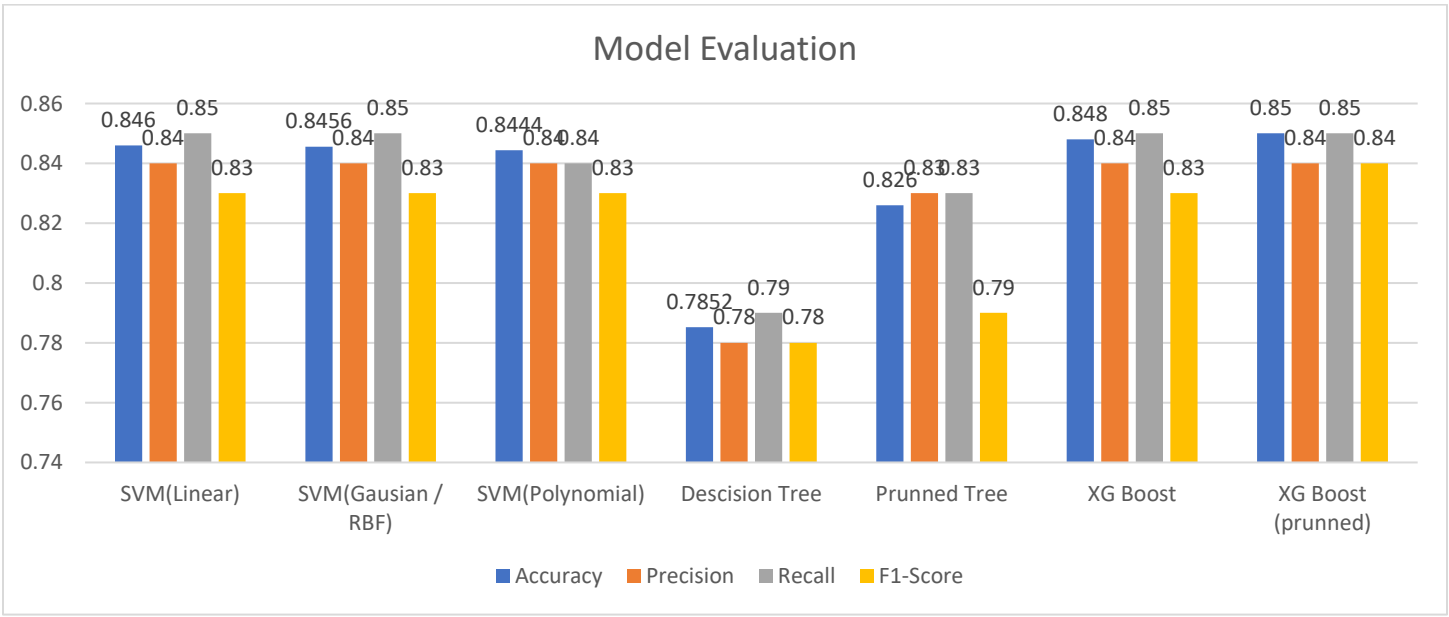
- The best kernals is SVM(polynomial) with a train and test accuracy of 0.88 and 0.84. It has a minor difference with the other kernals (linear, rbf)





Model Evaluation:

Algorithm	Accuracy	Precision	Recall	F1-Score
SVM(Linear)	0.846	0.84	0.85	0.83
SVM(Gaussian / RBF)	0.8456	0.84	0.85	0.83
SVM(Polynomial)	0.8444	0.84	0.84	0.83
Decision Tree	0.7852	0.78	0.79	0.78
Pruned Tree	0.826	0.83	0.83	0.79
XG Boost	0.848	0.84	0.85	0.83
XG Boost (pruned)	0.85	0.84	0.85	0.84



- XG Boost (pruned) has given the best accuracy, precision, recall and f1-score. SVM (Linear) and SVM (Gaussian/RBF) are almost close to XG Boost (pruned). Pruned tree gives better train and test error rate with comparison with other models.
- I haven't used the full dataset (100,000+) instead I used a subset of 10,000 out of the full dataset. I would get better result if all the instances are included in model. I omitted some variable cloud9am, cloud3pm. Including those values and filling the missing values with mean or median will yield better result.
- We can extract a variable called month from date column, that will give us better prediction.
- **Cross validation:** This significantly reduces bias as we are using most of the data for fitting and significantly reduces variance most of the data the data is also being used in validation set.

Conclusion:

1. XG Boost (pruned) performed better for both the data sets. It has better accuracy, precision and recall.
2. Decision tree performed really bad with the weather prediction dataset while pruned tree performed the worst for the gpu_runtime dataset.
3. Better results can be obtained if all the data points will be used in the model and adding relevant variables, extract new variable and implement hyperparameter tuning with wide range and 10 fold cross validation with 3 repeats.