Dataset 1 SGEMM GPU KERNEL PERFORMANCE

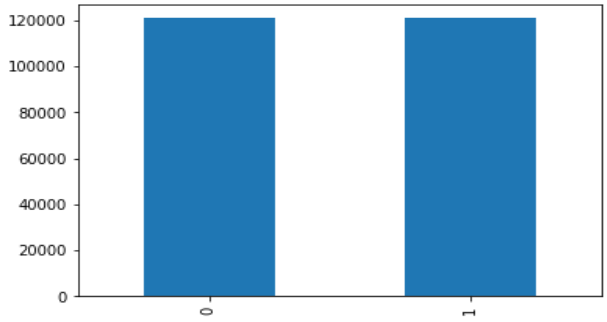
DATASET DESCRIPTION AND SUMMARY STATISTICS

This data set measures the running time of a matrix-matrix product A*B = C, where all matrices have size 2048 x 2048, using a parameterizable SGEMM GPU kernel with 241600 possible parameter combinations.

A brief description of the data set is as following:

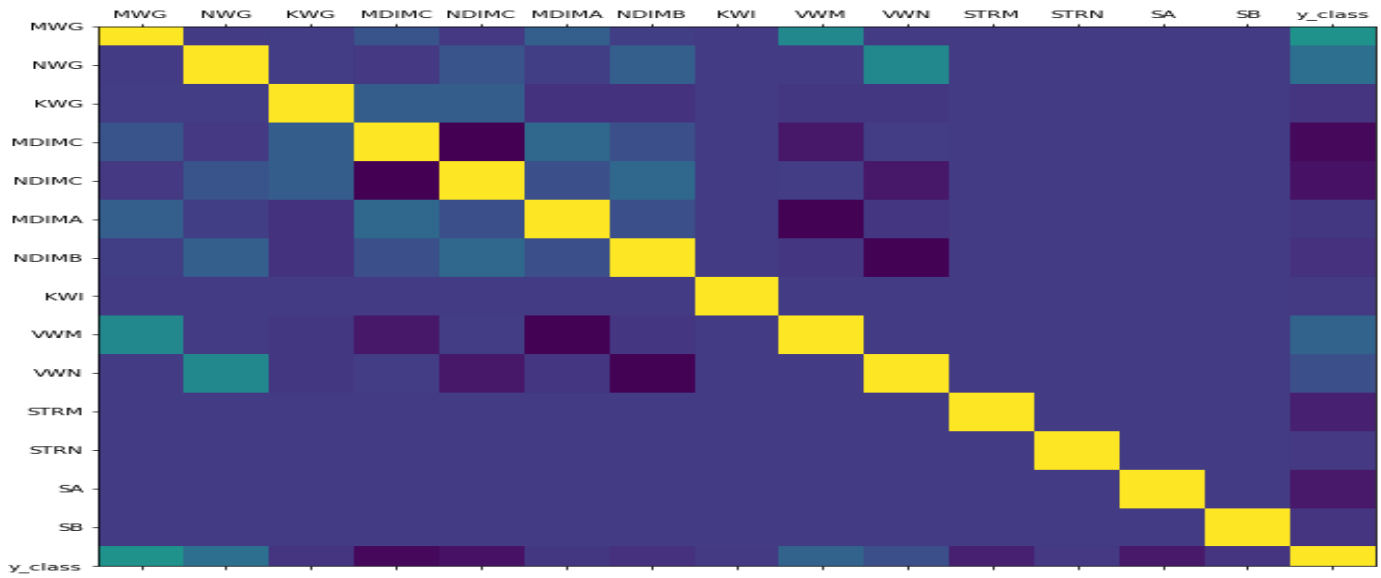
- Dataset consists of 241600 observations with 18 variables.
- For the training dataset, there are 14 parameters where the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary.
- The dependent variable for the linear regression model is Runs_Average: Average of 4 runs (the last 4 columns in the data set).
- No missing values are found in any column, so no missing value imputation is required.

From the figure above it is evident that the dataset is balanced as there are approximately equal number of 0's and 1's.



CORRELATION MATRIX

After checking the correlation between the data, it is observed that yellow colour represents max correlation and blue colour represents min correlation. We can see none of variable have correlation with any other variables. So, no problem of multicollinearity.



PART 2: SUPPORT VECTOR MACHINE

The SVM model performance can be improved by tuning parameter values. Some important parameter values that have an impact on the model performance are:

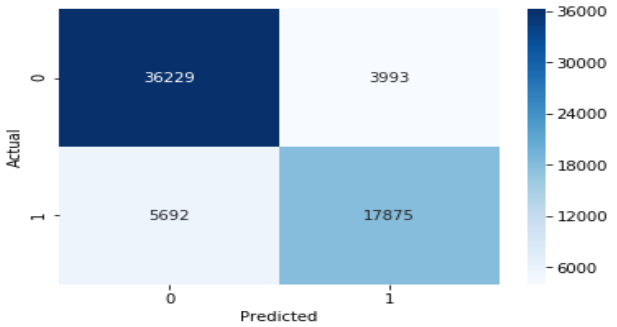
1. **kernel** – Various options are available such as linear, rbf, poly. The default used is rbf.
2. **gamma** - Kernel coefficient for 'rbf', 'poly' and 'sigmoid'. Higher the value of gamma, will try to exact fit the as per training data set i.e. generalization error and cause over-fitting problem.
3. **C** - Penalty parameter C of the error term. It also controls the tradeoff between smooth decision boundary and classifying the training points correctly.

Dataset 1 SGEMM GPU KERNEL PERFORMANCE

Splitting data in 70-30 ratio, training model on 70% and measuring accuracy on 30% test set.

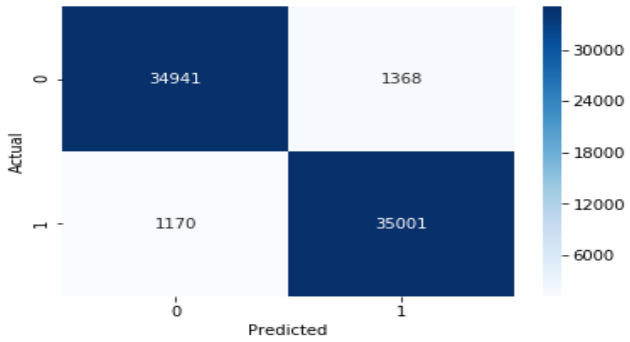
Linear Kernel: Accuracy = 84.81%

The number of observations that are correctly classified are 54104 observations.



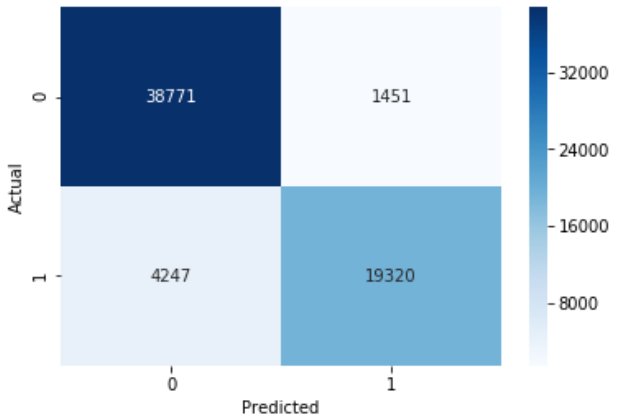
Gaussian(rbf) Kernel: Accuracy = 96.49%.

The number of observations that are correctly classified are 69942 observations.



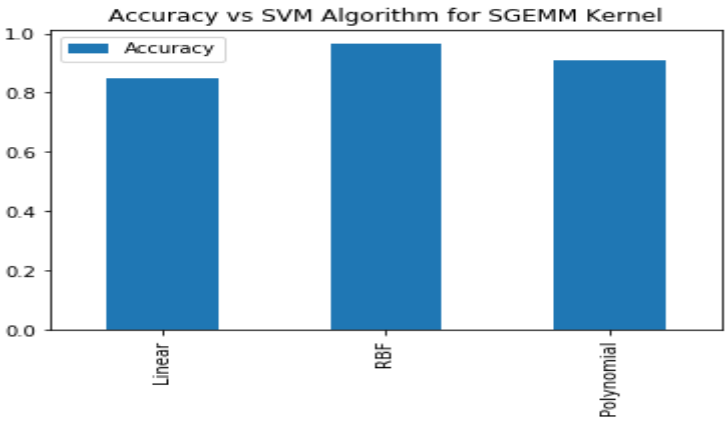
Polynomial Kernel: Accuracy = 91.06%

The number of observations classified correctly are 58091.



Comparison between Kernels:

From the figure above, it is evident that SVM with RBF kernel gave maximum accuracy (96.49%) in this dataset.

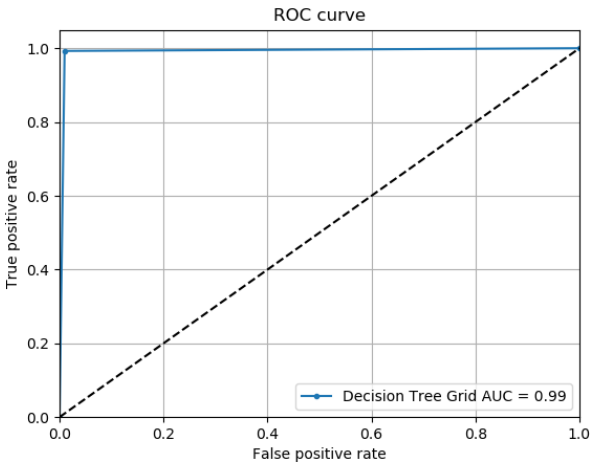


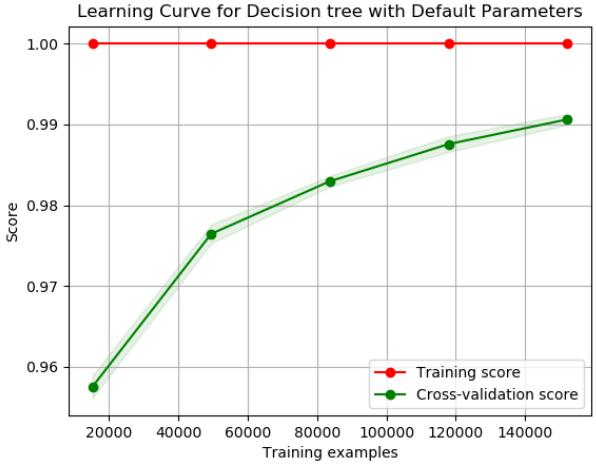
PART 3: DECISION TREE

Attribute selection is used for selecting the splitting criterion that partitions the data into the best possible manner. **Information gain, Gini index, max depth, splitter are some of the popular selection measures.**

Dataset 1 SGEMM GPU KERNEL PERFORMANCE

Decision Tree algorithm with default parameters gave an accuracy **99%.**



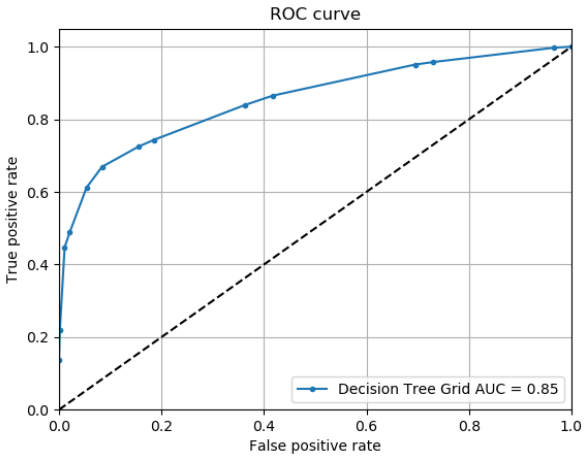


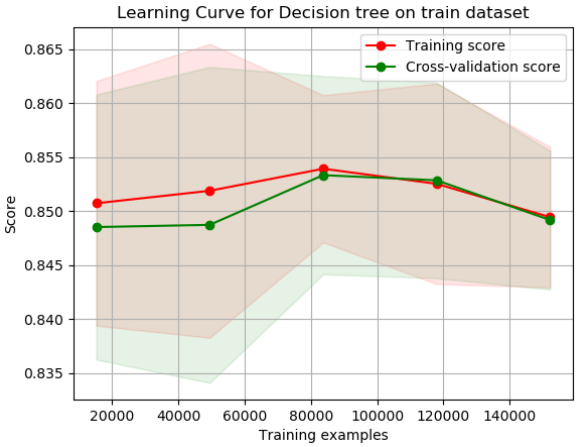
Here AUC = 1 which shows that it is performing well as it can distinguish between two classes easily. This is too good to be true case which shows that this is the case of overfitting as it is not learning much.

Training score is at its maximum regardless of the training examples. This indicates severe overfitting. Cross validation score increases with increase in dataset. The huge gap between the cross-validation score and training score indicates high variance. The complexity of the model needs to be reduced & hence pruning is required.

Decision Tree with GridSearchCV: Accuracy = 81%

When grid search is used for pruning to find out best hyper parameters along with 10-fold cross validation accuracy of 81% is received.





AUC = 0.85 which shows that it is a good model as it can distinguish between two classes but it reduces the problem of overfitting.

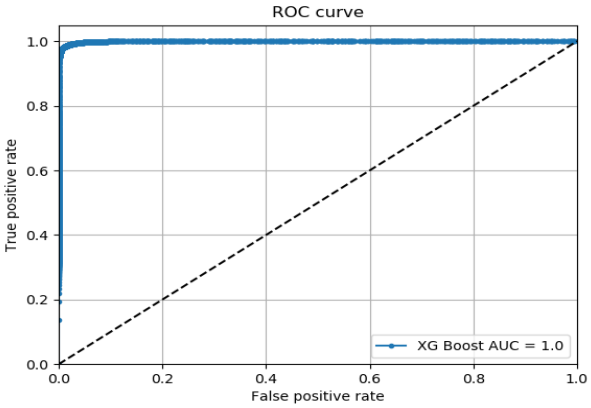
The learning curve above shows that there is a small gap between training & validation score indicating the model generalizes well with the increase in training size dataset the AUC score increases up to an extend and then decreases slightly.

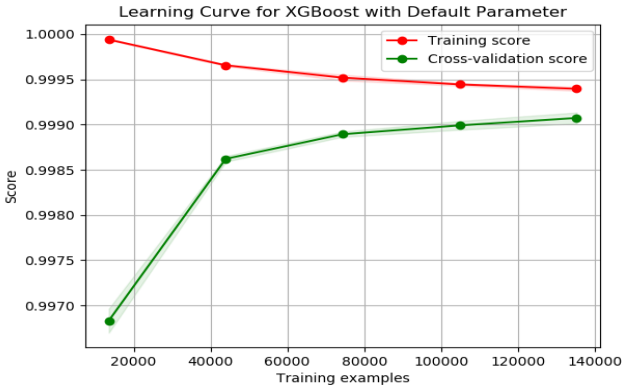
PART 4: GRADIENT BOOSTING

Dataset 1 SGEMM GPU KERNEL PERFORMANCE

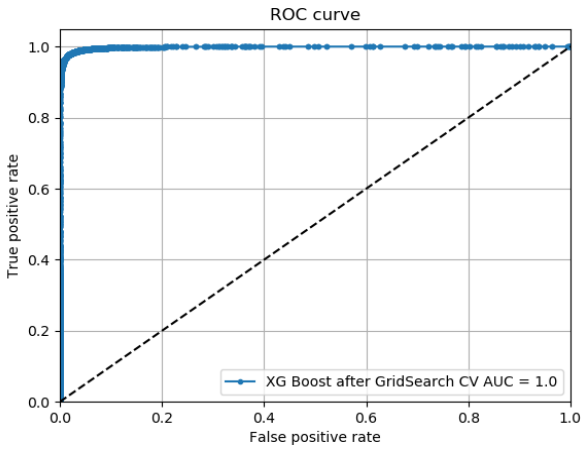
Boosting with default parameters

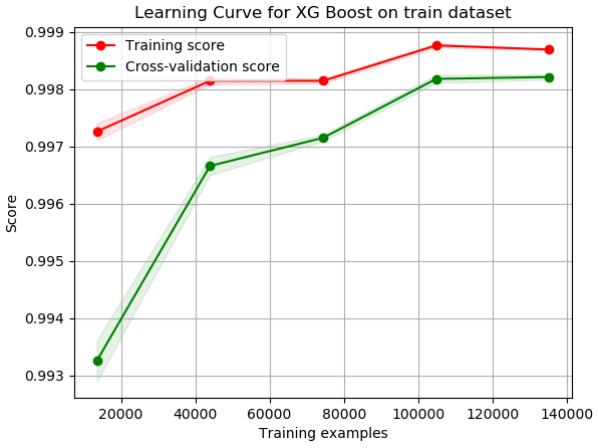
The learning curve above shows very high variance but with increase in training size the plots converges & variance decreases, probably with more data the algorithm would have performed much better. Also, the training score is higher than the validation score. So, there is a possibility of over-fitting. It has an accuracy of 98.46%





Boosting of Decision Tree (with GridSearchCV): - Accuracy = 98.78%





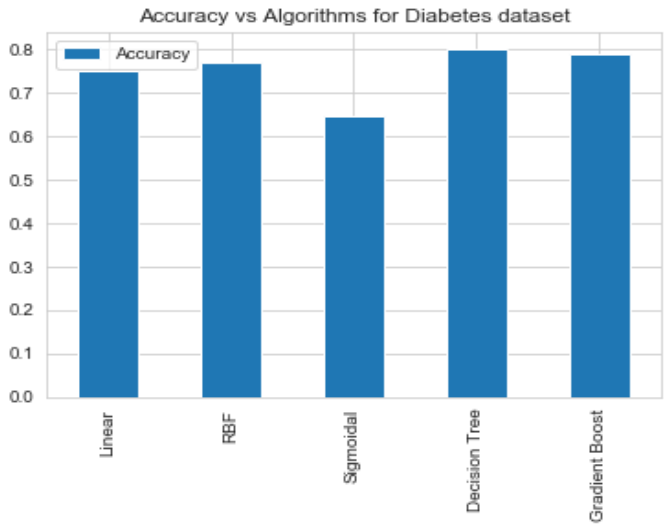
The learning curve above shows that the accuracy increases with increase in training data and variance descreaes. Thus the model generalizes better compared to model with default parameter.

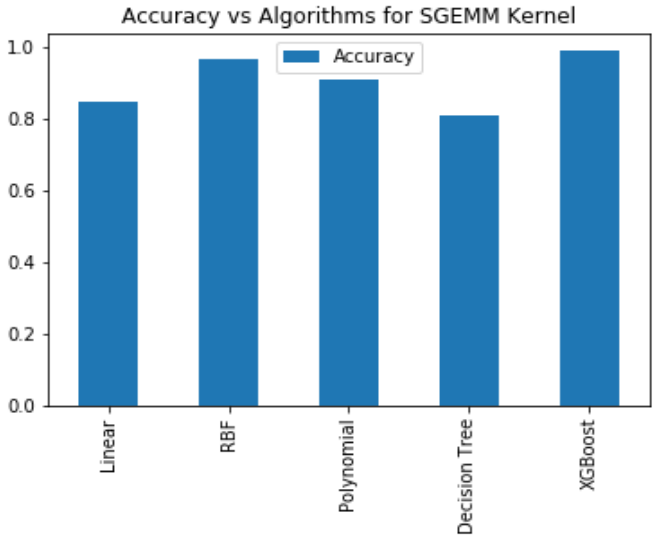
PART 5: CROSS VALIDATION

Dataset 1 SGEMM GPU KERNEL PERFORMANCE

I have performed all the algorithms by using grid search technique which includes cross validation as well.
The best result I found is with XGBoost having an accuracy of 98.78%.
Higher the AUC, better the model is at distinguished between classes. So, the best result is achieved by XGBoost with an AUC value of 0.9.

Model Comparison: - decision tree performed best in 1st dataset and in 2nd XGBoost performed best.





DISCUSSSIONS

Results: Machine learning model performance is relative and ideas of what score a good model can achieve only make sense and can only be interpreted in the context of the skill scores of other models also trained on the same data. So, for me the best results for SGEMM kernel performance 98.78% is the highest accuracy.

Cross validation: In general, there is a trade-off between accuracy and generalisation. The more accurate your classifier is on your training data, the less it will probably generalise (depends on your training data). This is called overfitting and cross validation tries to avoid that. Therefore, you mix training and test data on each fold. This method affected the results greatly in both the datasets. Like in SGEMM kernel performance when I implemented Decision tree, it clearly stated that it overfitted the training and testing set. To reduce it, I applied Grid search with cross validation which reduced the problem of overfitting in the dataset.

Pruning: I performed pruning by using grid search algorithm by setting the range of cost function and gamma and it showed after performing combinations which parameters are best for getting best results. By specifying max depth of the model, we can set when to stop further pruning. **Maximum depth refers to the length of the longest path from a root to a leaf.**

Kernel choice: I randomly choose kernels that performs the best amongst all other kernel. Also, the computational speed was another factor to choose the kernel. In SGEMM kernel performance, RBF performed the best.

Best performing algorithm: In SGEMM Kernel performance dataset, the best performing algorithm is XGBoost algorithm.