GPU RUN TIME PREDICTION

Introduction:

As we are marching towards more technological era, increase of data has increased manifold. With the increase in data, the need for faster processing and manipulation has also increased. In this linear and logistic regression analysis, we are computing the average run time for ideal combination of processors.

This analysis of ideal processors will come under supervised machine learning aiming to predict the fastest combination of processors .Techniques such as Gradient Descent algorithm, linear regression have been applied to get the best results.

Dataset:

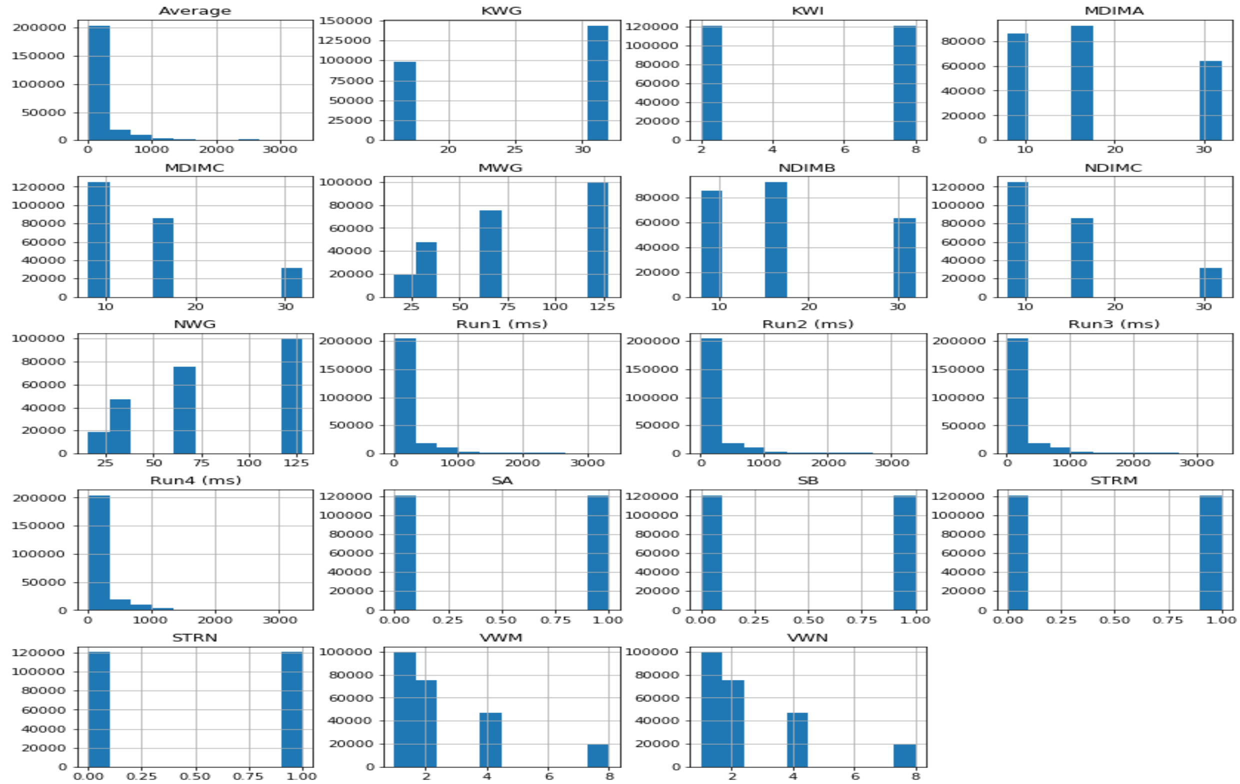
The dataset is download at:

https://archive.ics.uci.edu/ml/datasets/SGEMM+GPU+kernel+performance

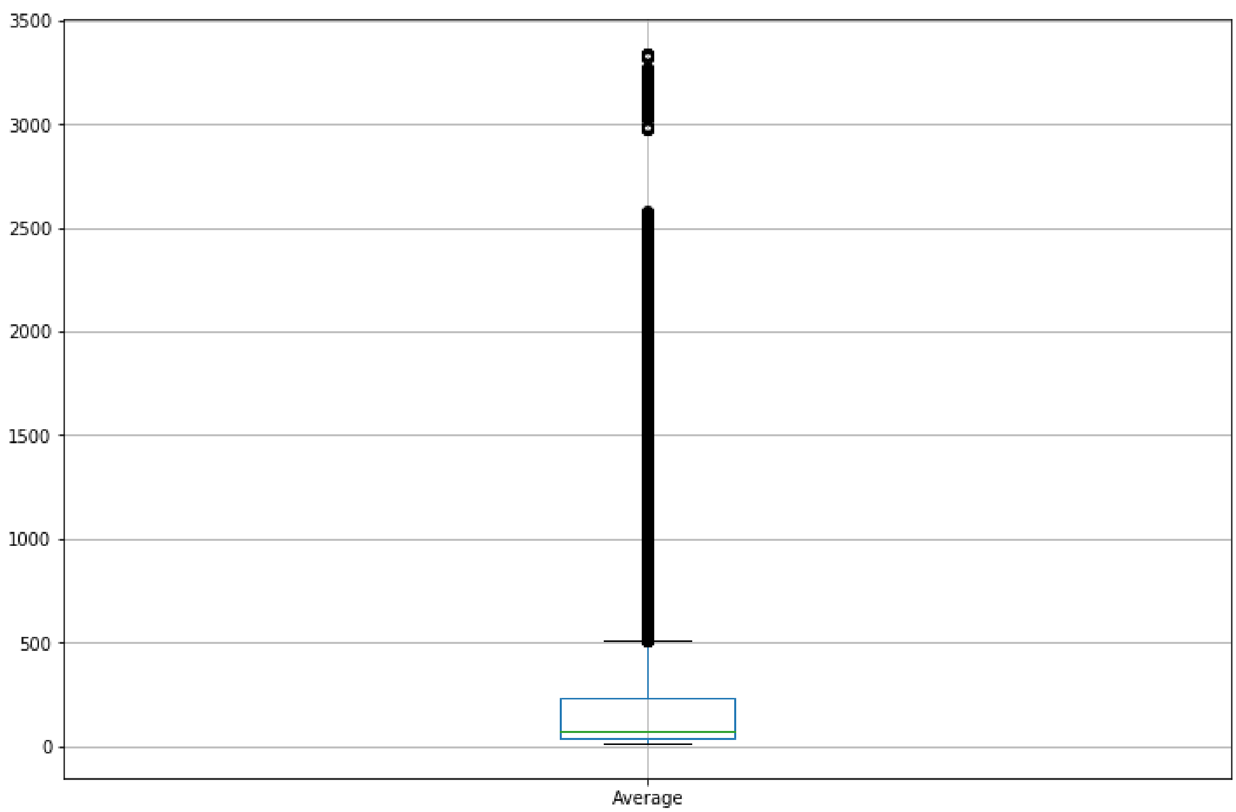
There are 14 parameter, the first 10 are ordinal and can only take up to 4 different powers of two values, and the 4 last variables are binary. Out of 1327104 total parameter combinations, only 241600 are feasible (due to various kernel constraints). This data set contains the results for all these feasible combinations. In this data set we don't have any null values present. Please find the screenshot below:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 241600 entries, 0 to 241599
Data columns (total 18 columns):
MWG          241600 non-null int64
NWG          241600 non-null int64
KWG          241600 non-null int64
MDIMC        241600 non-null int64
NDIMC        241600 non-null int64
MDIMA        241600 non-null int64
NDIMB        241600 non-null int64
KWI          241600 non-null int64
VWM          241600 non-null int64
VWN          241600 non-null int64
STRM         241600 non-null int64
STRN         241600 non-null int64
SA           241600 non-null int64
SB           241600 non-null int64
Run1 (ms)    241600 non-null float64
Run2 (ms)    241600 non-null float64
Run3 (ms)    241600 non-null float64
Run4 (ms)    241600 non-null float64
dtypes: float64(4), int64(14)
memory usage: 33.2 MB
```

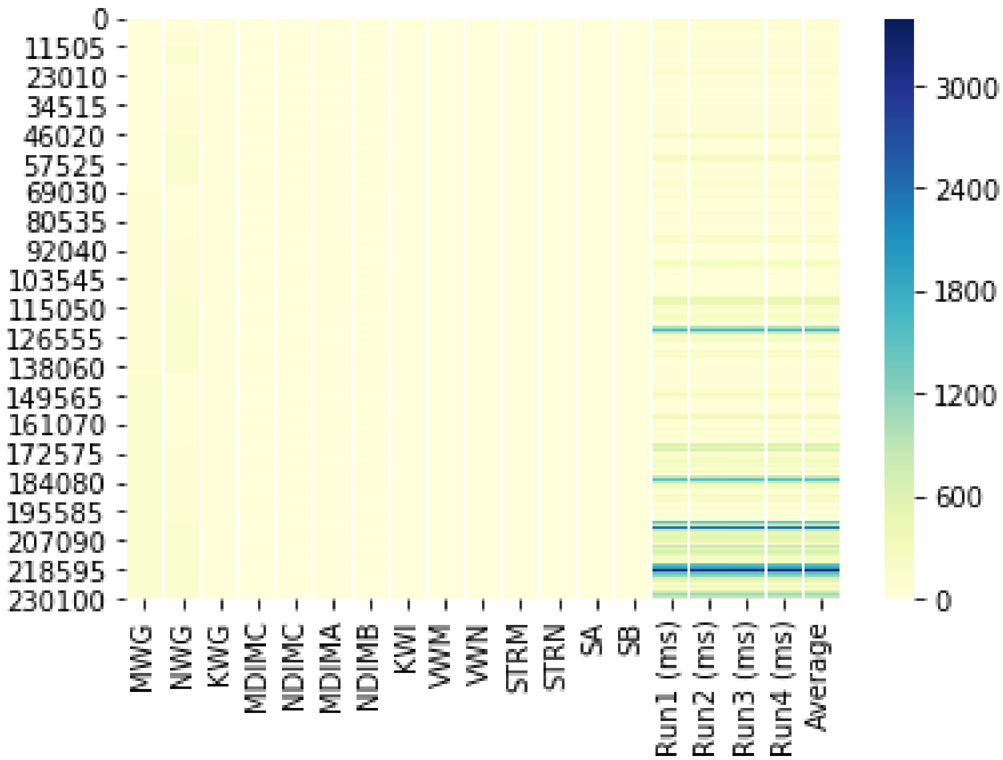
Feature distribution for the above variables is described below:



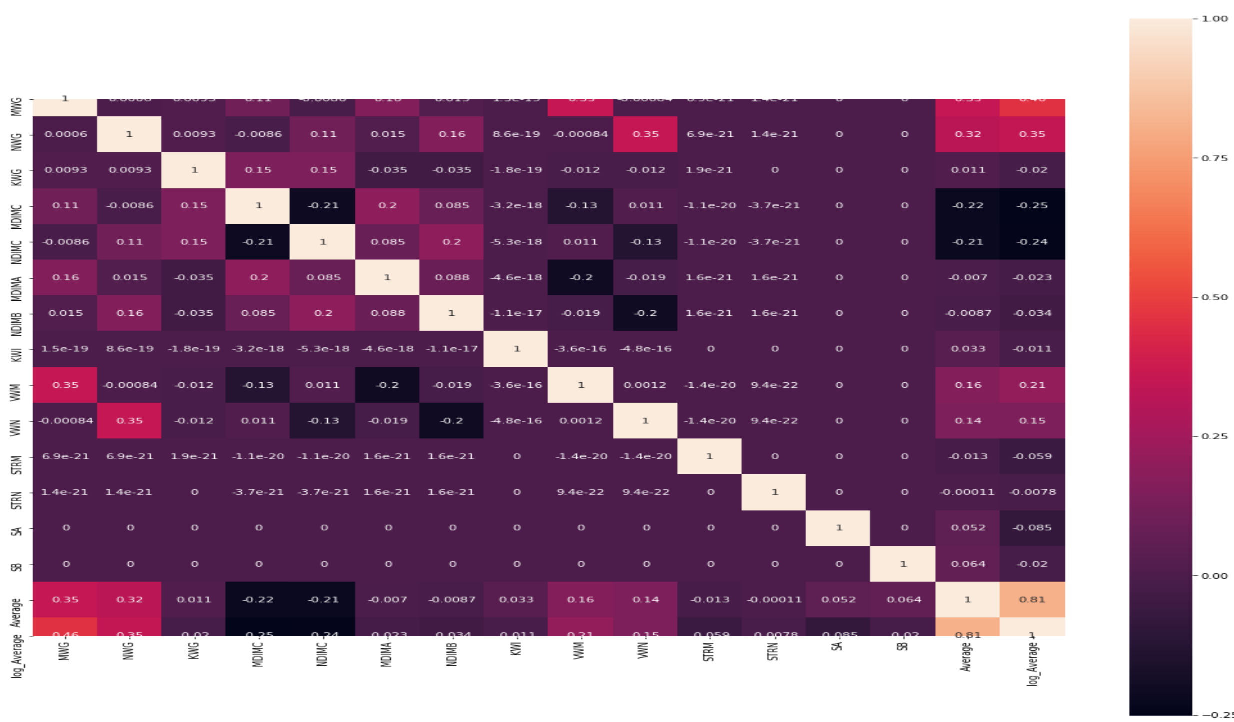
Boxplot Application for the above features:



Heatmap for the above features:



Correlation matrix



Part 1: Download the dataset and partition it randomly into train and test set using a good train/test split percentage.

I have randomly separated the data into testing and training sets using different percentages as 0.8 and 0.2 which means 80 percent of data will be used in training and 20 percent of data will be used to validate the model.

Part 2 :Design a linear regression model to model the average GPU run time. Include your regression model equation in the report.

Part 3: Implement the gradient descent algorithm with batch update rule. Use the same cost function as in the class (sum of squared error). Report your initial parameter values.

I have designed the linear regression model to model the average runtime of processors.

The model equation is

Average=beta0 + beta1*x1+beta2*x2+beta3*x3…………………………….beta14*x4

I have built a custom implementation of the gradient descent function which implements the linear model. The initial parameters that I have got are

Features	coefficients
X1	218.0101525
X2	141.6565995
X3	130.84711401
X4	41.16106933
X5	-131.56267379
X6	-128.83840281
X7	10.12431447
X8	9.50365155
X9	12.39775437
X10	-2.77959938
X11	-6.34156052

X12	-4.59608693
X13	0.27944157
X14	19.46804351
X15	23.84465519

Part 4:

Part 4: Convert this problem into a binary classification problem. The target variable should have two categories. Implement logistic regression to carry out classification on this data set. Report accuracy/error metrics for train and test sets.

I have converted this problem into binary classification problem using the Average column in the dataset and have converted the target variable into 0 and 1. I have taken the mean of log_Average column and have classified the average into two categories that is above mean and below mean as 1 and 0 respectively. Accuracy my model has achieved is 89.9 percent.

EXPERIMENTS:

Experiment 1:

Experiment with various parameters for linear and logistic regression (e.g. learning rate ∝) and report on your findings as how the error/accuracy varies for train and test sets with varying these parameters. Plot the results. Report the best values of the parameters.

1. Linear Regression
   I have fixed the threshold at 0.0000001 and have varied learning rate [1,0.8, 0.6, 0.4, 0.2, 0.1, 0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001].
   To find the best value of alpha , I tried the different combinations of learning rate
   Following learning rates were the last batch of learning rates where I got my local minima: [0.00098,0.00096,0.00094,0.00093,0.00092,0.00091,0.00090,0.00089,0.00088,0.00087]
   I got the minimum RMSE at 0.0009 and that was the best alpha value for the given threshold.

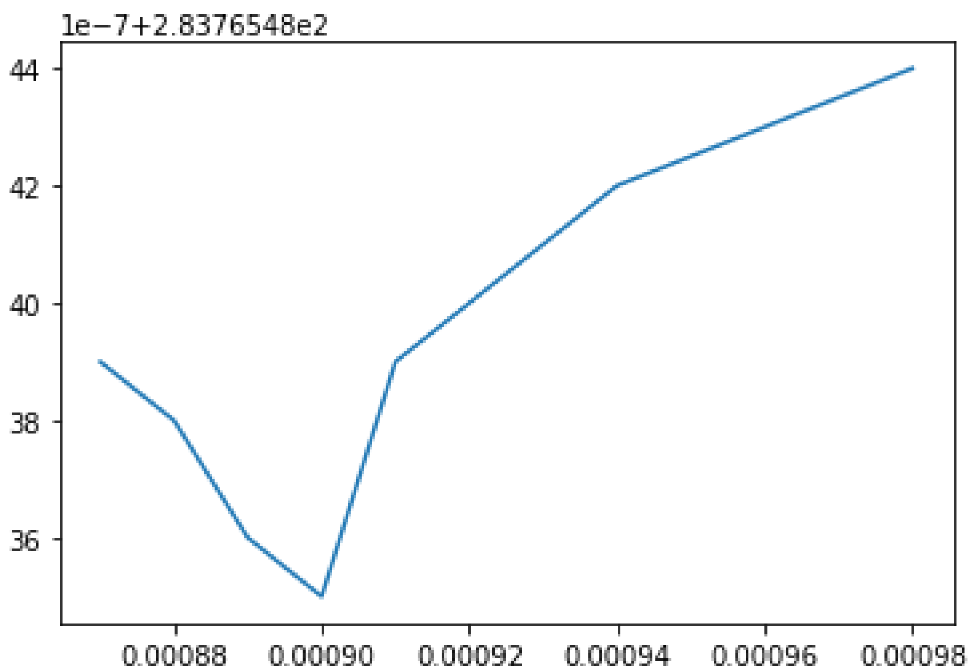
   Graphs for Learning Rate:

Learning Rate	RMSE
0.00098	283.7654844
0.00096	283.7654843
0.00094	283.7654842

0.00093	283.7654841
0.00092	283.7654840
0.00091	283.7654839
0.00090	283.7654835
0.00089	283.7654836
0.00088	283.7654838
0.00087	283.7654839

283.7654844,283.7654843,283.7654842,283.7654841,283.7654840,283.7654839,283.7654835,283.7654836,283.7654838,283.7654839]

The final graph for RMSE and Alpha is below:



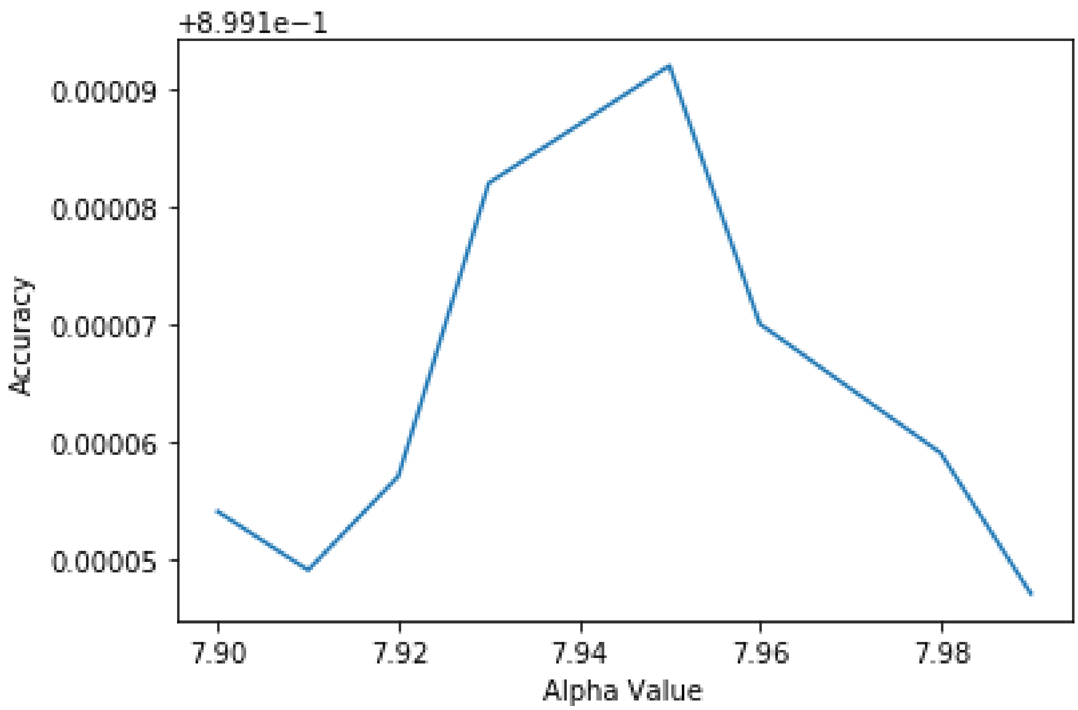
2. Logistic Regression
I have fixed threshold at 0.0000001 and have varied learning rate [0.0001 ,0.001 ,0.005, 0.01, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1].
To find the best value of alpha, I have tried the different combinations of alpha. Following were the last batch of alpha. []. I got the minimum RMSE at ------.and that was the best value for the given threshold.

Graphs for Different learning rate:

Learning Rate	Accuracy
7.9	0.899154
7.91	0.899149
7.92	0.899157
7.93	0.899182

7.95	0.899192
7.96	0.89917
7.98	0.899159
7.99	0.899147

Graph for Logistic:



EXPERIMENT 2

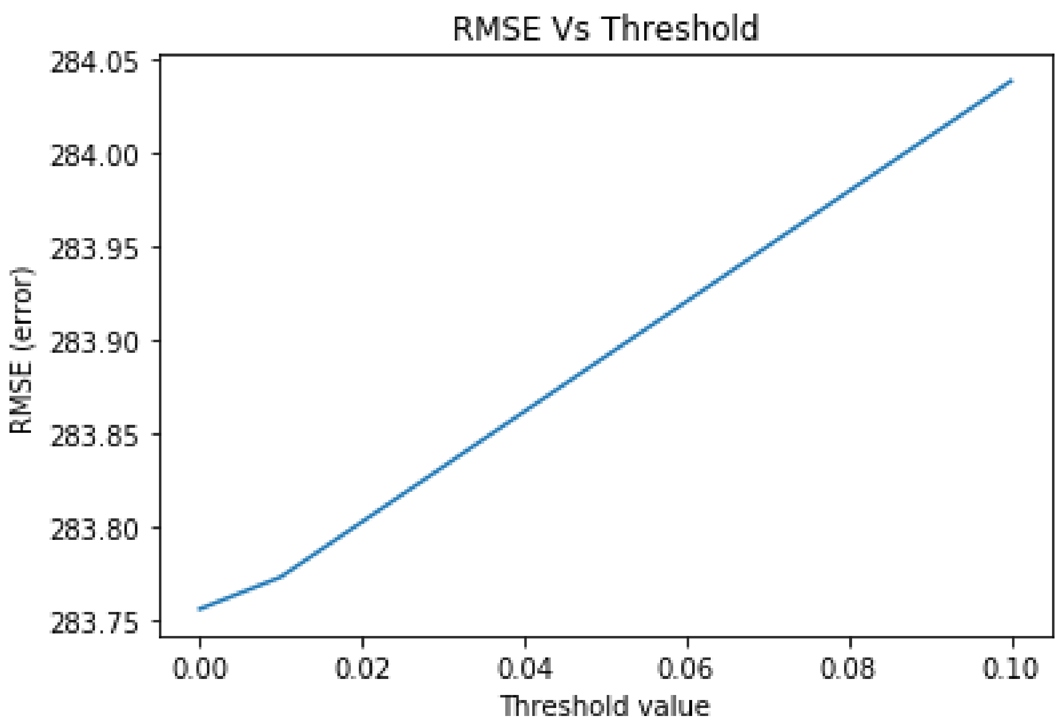
Experiment with various thresholds for convergence for linear and logistic regression. Plot error results for train and test sets as a function of threshold and describe how varying the threshold affects error. Pick your best threshold and plot train and test error (in one figure) as a function of number of gradient descent iterations.

For Linear Regression

Changing Threshold values:

Threshold Level	RMSE	Learning Rate
0.00001	283.75648345	0.0009
0.01	283.77357268	0.0009
0.1	284.03815423	0.0009

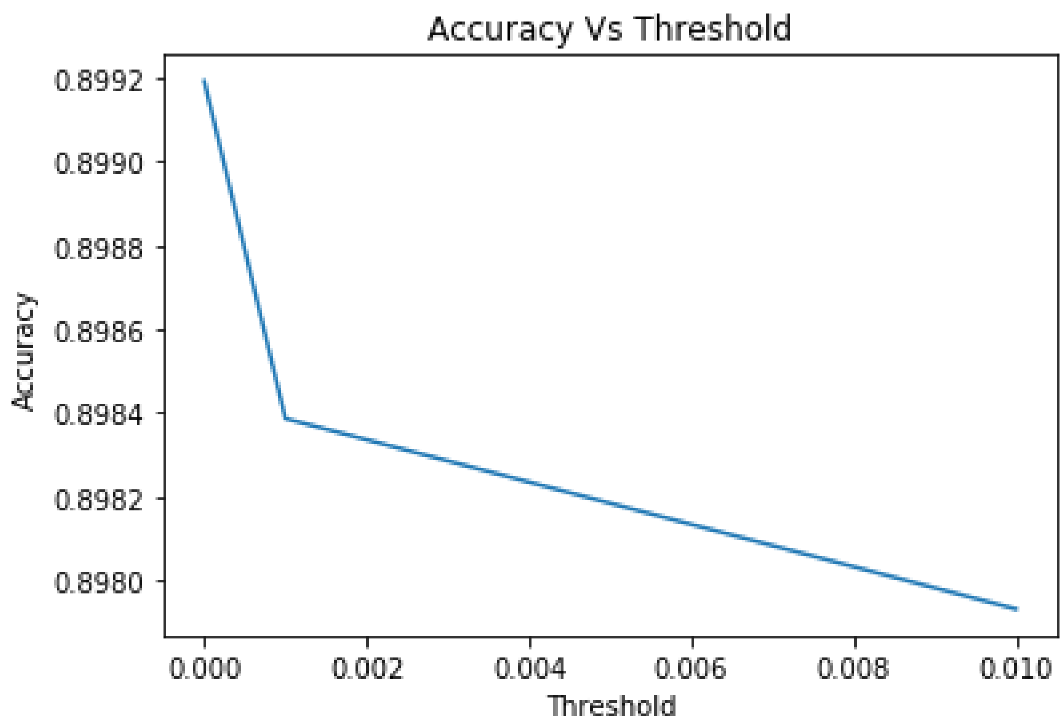
GRAPHS FOR LINEAR REGRESSION



For Logistic Regression

Threshold Level	Learning Rate	Accuracy
0.0000001	7.95	0.899192
0.001	7.95	0.898385761589404
0.01	7.95	0.8979304635761589

GRAPHS FOR LOGISTIC REGRESSION



EXPERIMENT 3

Pick eight features randomly and retrain your models only on these ten features. Compare train and test error results for the case of using your original set of features (14) and eight random features. Report the ten randomly selected features.

I have randomly taken 8 features and have modeled my linear and logistic models on those features:

8 Features: ["MWG","NWG","KWG","MDIMC","NDIMC","MDIMA","NDIMB","KWI"]

Target Variable: [“Average]

1) Linear Model

	RMSE	Cost Value	Iterations
Model – 8 Random variables	285.54262715	40889.5117525077	4747
Best Model using Experiment 1 and 2	284.03815423	40395.446610390405	5200

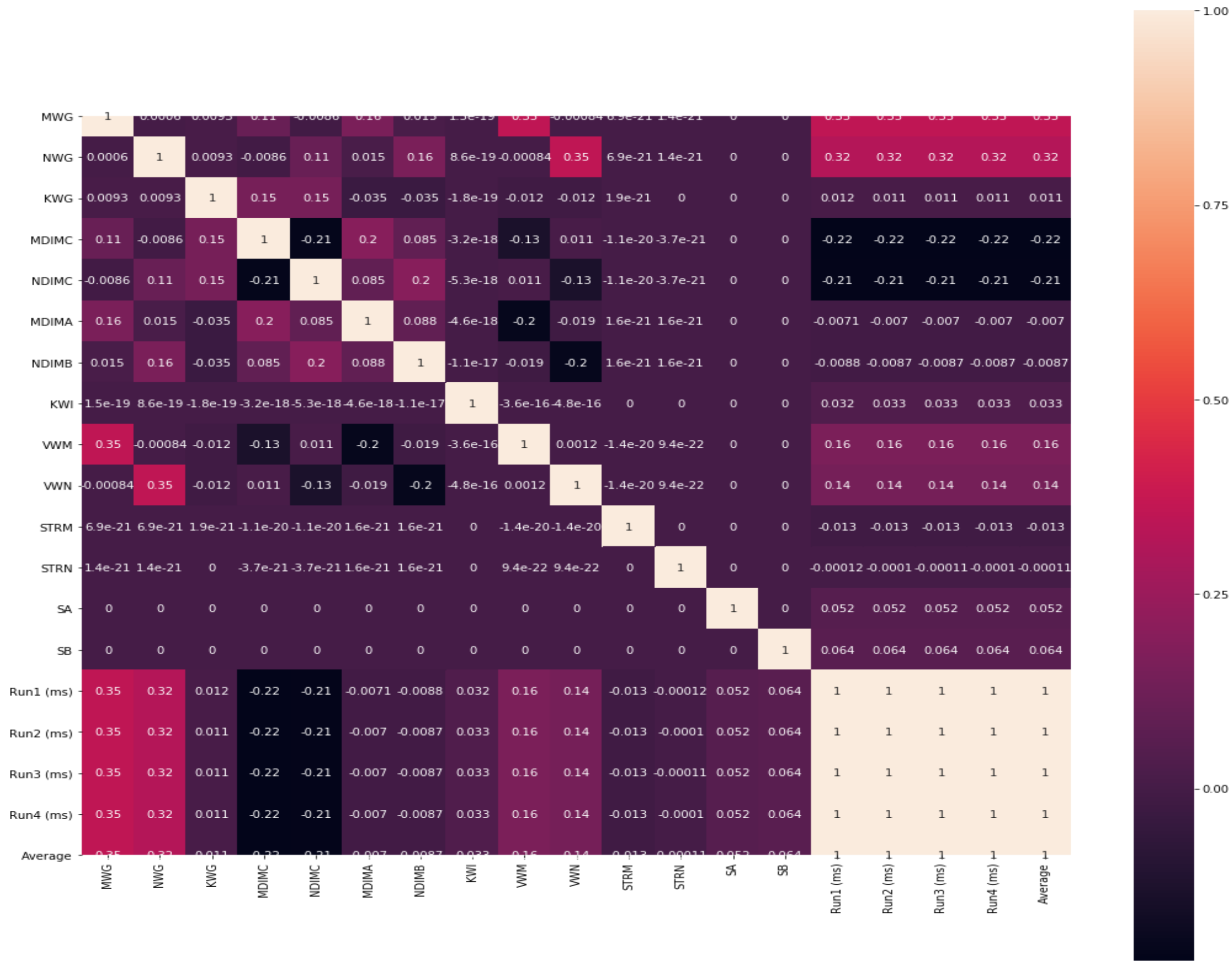
2) Logistic Model

	Accuracy	Cost Value	Iterations
Model-8 Random Variables	0.9218129139072848	0.35384131313366624	10000
Best Model Experiment 1 and 2	0.899192	40395.446	10000

EXPERIMENT 4

Now pick eight features that you think are best suited to predict the output, and retrain your models using these ten features. Compare to the case of using your original set of features and to the random features case. Did your choice of features provide better results than picking random features? Why? Did your choice of features provide better results than using all features? Why?

According to the correlation matrix described below:



I have taken the best 8 features which are highly co related to the Average i.e the target variable:

The variables I have taken are listed below:

['MWG', 'NWG', 'KWG', 'MDIMC', 'NDIMC', 'VWM', 'SA','VWN']

1) Linear Model

Models	RMSE	Cost Value	Iterations
Randomly selected	285.542627156	40889.5117525077	4747
Selected 8 features	285.542627154	40854.0281642906	5191

When I have taken the random 8 variables and selected 8 features, there is just slight difference between the RMSE between the two models. Cost Value for the selected eight features is less as compared to the randomly selected feature model.

2) Logistic Model

Models	Accuracy	Cost Value
Randomly selected	0.9218129139072848	0.35384131313366624
Selected 8 features	0.8997102649006623	0.34832213626441216

For logistic Regression, the randomly selected features gave me better resutts as compared to the best selected 8 features.

Discussion :

Describe your interpretation of the results. What do you think matters the most for predicting the GPU run time? What other steps you could have taken with regards to modeling to get better results?

Answer: The pre processing and analysis of the data mattered the most while predicting results through our model.

We could have used PCA (Principal component analysis) to determine which features are most relevant ones. This could have saved us time and gave us better understanding.

We could have used better classification techniques such as KNN, to classify results better.