**Implementation of Linear and Logistic Regression Using Gradient Descent Algorithm**

**Dataset description and goal:**

The given data set contains information on SGEMM GPU Kernel Performance which captures the running time of a matrix-matrix product operation. The goal of the project is to implement linear and logistic regression on the given dataset using gradient descent algorithm with batch update, predict the average GPU run time, report the optimal parameter values, and perform feature selection to get the best model. The dataset information can be viewed and downloaded at UCI Machine learning repository.

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

On quick examination, the following characteristics are observed from the dataset.

* There are 18 columns and 241600 rows
* There are no missing values
* The target variable for the model is average of the four GPU run times
* The independent variables are not highly correlated. Hence there is no necessity to drop any variables from the model.
* There are few outliers present, and they need to be removed before running regression and feature selection.

**Assumptions and initial dataset preparation:**

The following assumptions and initial operations are made on the dataset for the implementation of linear and logistic regression.

* The columns 1-10 and 15-18 are considered continuous and columns 11-14 are considered categorical dummy variables.
* STRM interacts with MDIMA, STRN interacts with NDIMB, SA interacts with MWG, and SB interacts with NWG
* The average of columns 15-18 is calculated as target variable and a new dataset has been created with 15 columns and 241600 rows.
* Scaling has been performed on new dataset for columns 1-10 by subtracting the minimum and dividing by range.
* For logistic regression, column 15 is converted to either 1 or 0 using its mean value as cut off.
* The number of outliers (above or below 3 standard deviations from mean) is 5251 (accounts for 2.17 % of data) and they have been removed from the given dataset.

**Tasks 1,2,3,4:**

The dataset is scaled and partitioned into train set with 70% data and test set with 30% data. A function generates 15 values at random to get the initial betas for both linear and logistic regressions. (The number of columns is 14, hence we need 15 betas for full model).

The values are: β0 = 0.0523636, β1 = 0.08718668, β2 = 0.40724176, β3 = 0.10770023, β4 = 0.90119888,

β5 = 0.03815367, β6 = 0.53620204, β7 = 0.3321977, β8 = 0.85208662, β9 = 0.1596624,

β10 = 0.33721666, β11 = 0.33379639, β12 = 0.24516335, β13 = 0.00167055, β14 = 0.43627579

Other parameters considered are:

* learning rate (alpha) = 0.001
* threshold of convergence = 0.000001
* maximum number of iterations = 10000
* cut off probability = 0.5 (for logistic regression; to convert predicted y probabilities to 0 or 1)
* cut off value = mean of target variable (for logistic regression; to convert actual y values to 0 or 1)

**Linear regression:**

The regression equation to model the given dataset is:

Run\_time = β0 + β1\*mwg + β2\*nwg + β3\*kwg + β4\*mdimc + β5\*ndimc + β6\*mdima + β7\*ndimb + β8\*kwi

+ β9\*vwm + β10\*vwn + β11\*(strm\* mdima) + β12\*(strn\*ndimb) + β13\*(sa\*mwg)

+ β14\*(sb\*nwg)

The optimal beta values are:

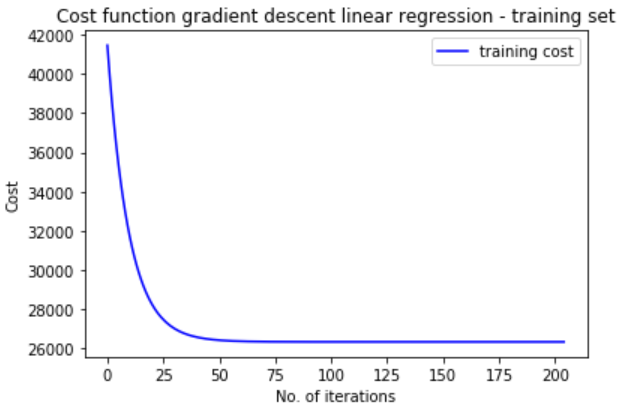
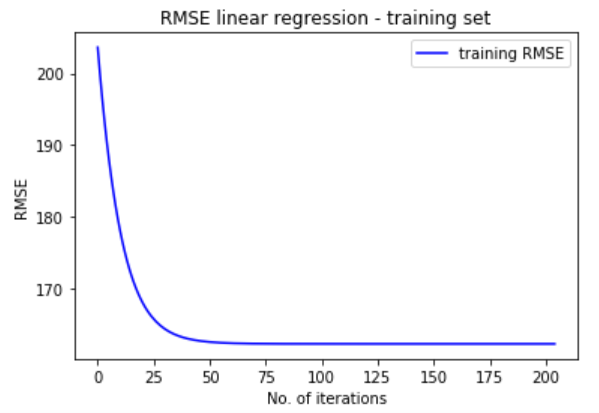
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 | β9 | β10 | β11 | β12 | β13 | β14 |
| 24.53 | 24.57 | 24.89 | 24.59 | 25.38 | 24.52 | 25.02 | 24.81 | 25.33 | 24.64 | 24.82 | 24.81 | 24.73 | 24.48 | 24.92 |

Minimum cost for training set is 26332.56

RMSE for training set is 162.27

Estimated test cost is 26268.56

Estimated test RMSE is 162.07

**Logistic regression:**

The regression equation to model the given dataset is:

Run\_time (0/1) = β0 + β1\*mwg + β2\*nwg + β3\*kwg + β4\*mdimc + β5\*ndimc + β6\*mdima + β7\*ndimb

+ β8\*kwi + β9\*vwm + β10\*vwn + β11\*(strm\* mdima) + β12\*(strn\*ndimb) + β13\*(sa\*mwg)

+ β14\*(sb\*nwg)

The optimal beta values are:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 | β9 | β10 | β11 | β12 | β13 | β14 |
| -0.78 | 0.279 | 0.45 | -0.418 | 0.24 | -0.39 | 0.111 | -0.03 | 0.149 | 0.231 | 0.316 | -0.106 | -0.14 | -0.46 | -0.16 |

Minimum Cost is: 0.5410 (For train set)

Accuracy of train set is : 0.726

Estimated test cost is 0.5407

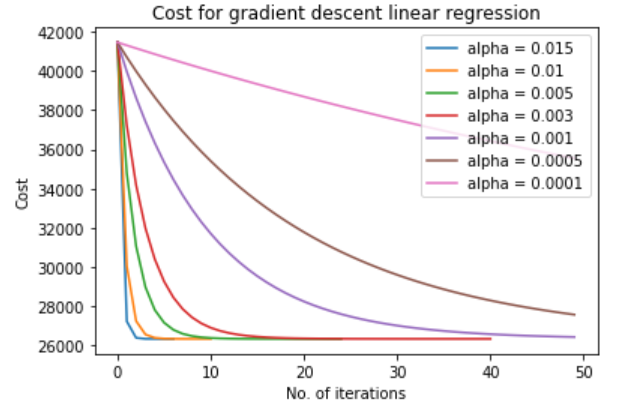
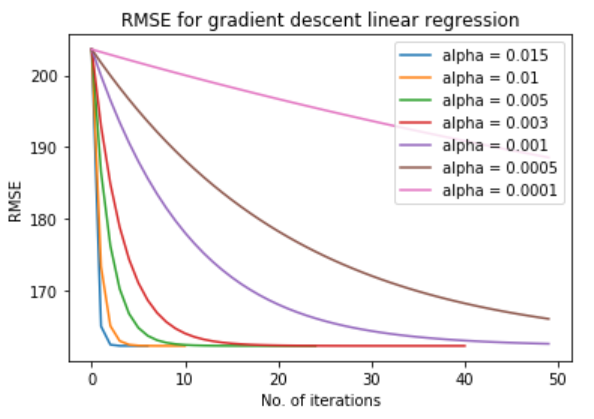
Accuracy of test set is : 0.7247

**Experiment 1:**

1. **Varying alpha:** The gradient descent algorithm is executed multiple times, each time with a different value for learning rate, keeping all other parameters constant.

*Linear Regression:*

The corresponding Cost Vs Iterations & RMSE Vs Iterations curves are plotted for training set, and RMSE value is calculated for test set. Maximum number of iterations is set to 50 to obtain good scaling for the output plot.

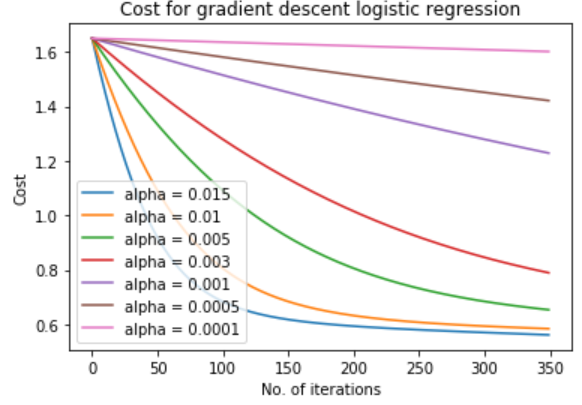
 

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Alpha** | **0.015** | **0.01** | **0.005** | **0.003** | **0.001** | **0.0005** | **0.0001** |
| **Train RMSE** | 162.2731 | 162.2731 | 162.2731 | 162.2732 | 162.5645 | 166.0935 | 188.5355 |
| **Test RMSE** | 162.0758 | 162.0757 | 162.0756 | 162.0755 | 162.3209 | 165.6004 | 188.0014 |
| **Time taken** | 0.78 | 0.92 | 1.34 | 1.09 | 1.17 | 1.16 | 0.98 |
| **Iterations** | 8 | 12 | 26 | 42 | 50 | 50 | 50 |

From the above graph and table, it is inferred that the number of iterations for convergence, RMSE values, and time taken for code execution vary for every alpha. A strange pattern is observed in the RMSE values of Train and Test. The values remain constant or decrease gradually until alpha = 0.003, and then start to increase significantly. Thus, for linear regression, the ***optimal alpha value should be around 0.003*** to obtain best results.

*Logistic Regression:*

The corresponding Cost Vs Iterations curve for training set and Accuracy values for test set are calculated. Maximum number of iterations is set to 350 to obtain good scaling for the output plot.



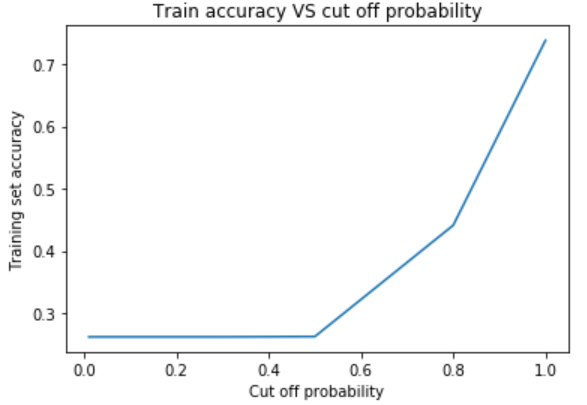
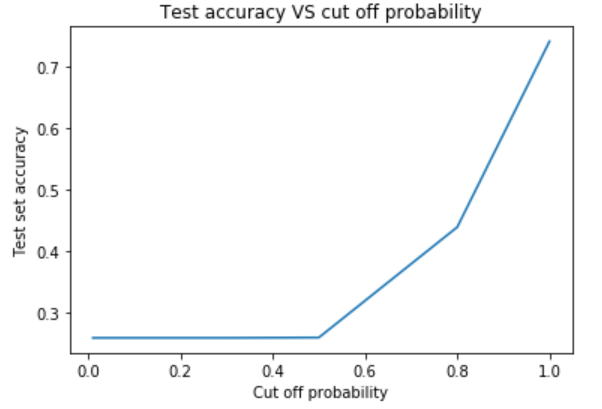
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Alpha** | **20** | **10** | **5** | **3** | **2.5** | **2** | **1.5** | **1** | **0.1** |
| **Test Accuracy** | 0.7058 | 0.7058 | 0.3201 | 0.7109 | 0.9210 | 0.9209 | 0.9207 | 0.9204 | 0.916 |
| **Time taken** | 1.58 | 1.89 | 1.61 | 1.53 | 11.81 | 14.78 | 16.42 | 21.8 | 55.6 |
| **Iterations** | 5 | 5 | 4 | 7 | 642 | 740 | 885 | 1130 | 3758 |

The following table shows metrics for various alpha, considering maximum number of iterations as 10000, and keeping all other parameters constant.

From the table, it is seen that the test accuracy increases with decrease in alpha, peaks at alpha = 2.5, and then starts to decrease. Hence for logistic regression, ***optimal alpha value should be around 2.5*** to obtain best results.

1. **Varying cut off probability:** The gradient descent algorithm for logistic regression is executed multiple times, each time with a different value for cut off probability, keeping all other parameters constant. The corresponding Accuracy Vs Cut off probability curve is plotted.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Cut off prob.** | **1** | **0.8** | **0.5** | **0.3** | **0.1** | **0.01** |
| **Train Accuracy** | 0.706 | 0.448 | 0.2939 | 0.293 | 0.293 | 0.293 |
| **Test Accuracy** | 0.705 | 0.445 | 0.2946 | 0.294 | 0.294 | 0.294 |

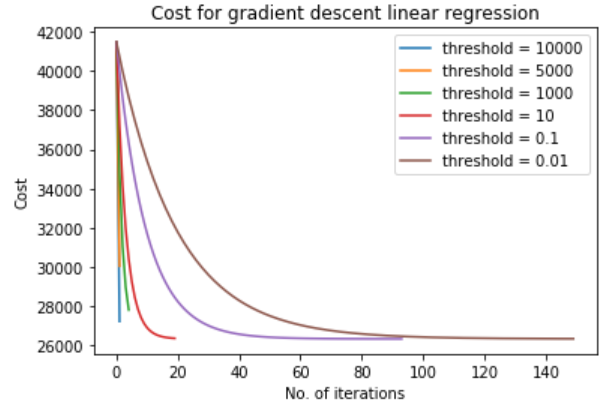
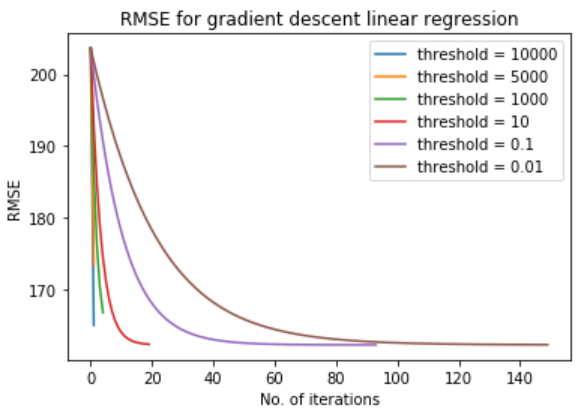
From the above graph and table, it is inferred that the accuracy increases steeply with increase in cut off probability. The ***cut off probability should be around 1*** to obtain best results.

**Experiment 2:**

**Varying threshold of convergence:** The gradient descent algorithm is executed multiple times, each time with a different value for threshold of convergence, keeping all other parameters constant.

*Linear Regression:*

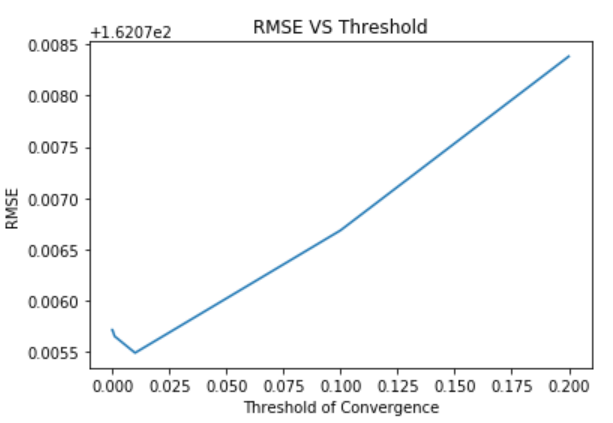
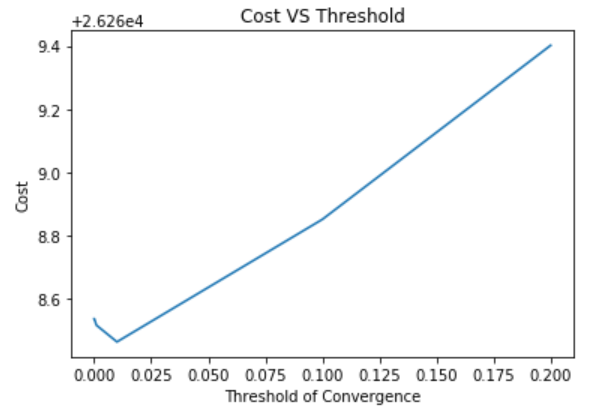
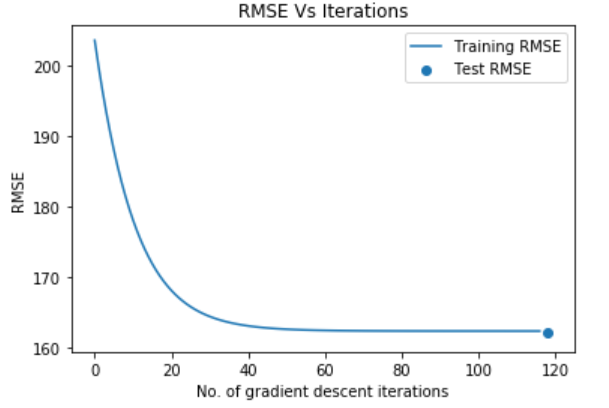
The corresponding Cost Vs Iterations & RMSE Vs Iterations curves are plotted for training set, and RMSE value is calculated for test set. For the below 2 plots, alpha is also varied to observe the difference in the convergence thresholds (Else the curves overlap). When threshold is high, the convergence to global minimum is not obtained. Maximum number of iterations is set to 150 to obtain good scaling for the output plot.

The following table presents the statistics for different thresholds, keeping all other parameters constant. We see that the test RMSE decreases significantly until threshold = 0.01 and then starts to increase again.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Threshold** | **10000** | **1000** | **100** | **10** | **1** | **0.5** | **0.1** | **0.01** | **0.001** | **0.0001** | **0.00001** |
| **Train RMSE** | 199.92 | 190.59 | 165.09 | 162.56 | 162.30 | 162.28 | 162.276 | 162.273 | 162.273 | 162.273 | 162.273 |
| **Test RMSE** | 196.25 | 187.71 | 164.57 | 162.32 | 162.09 | 162.08 | 162.076 | 162.0754 | 162.0756 | 162.0757 | 162.0758 |
| **Iterations** | 3 | 6 | 29 | 51 | 73 | 80 | 95 | 118 | 140 | 162 | 184 |
| **Time** | 0.80 | 0.91 | 1.30 | 1.39 | 1.53 | 1.43 | 1.69 | 2.06 | 1.93 | 2.24 | 2.04 |

Following plots 1 and 2 show how Cost and RMSE for test set varies with respect to threshold. The minimum cost and RMSE is obtained when threshold = 0.01. Hence, ***the optimal value of threshold is 0.01.***

The 3rd plot shows the train error as a function of number of gradient descent iterations when threshold = 0.01. The test error is a single predicted value, 162.075, which is obtained from the optimal beta values in training set.

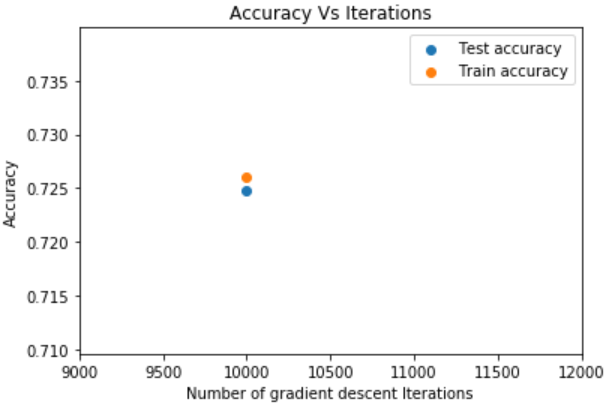
*Logistic Regression:*

For different thresholds, the corresponding accuracy value for train and test set is calculated.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Threshold** | **0.01** | **0.005** | **0.001** | **0.0005** | **0.0001** | **5e-05** | **1e-05** | **1e-06** | **1e-07** | **1e-08** |
| **Train Accuracy** | 0.293 | 0.293 | 0.294 | 0.3443 | 0.573 | 0.638 | 0.7119 | 0.7259 | 0.7259 | 0.7259 |
| **Test Accuracy** | 0.294 | 0.294 | 0.295 | 0.3445 | 0.572 | 0.639 | 0.7117 | 0.7247 | 0.7247 | 0.7247 |
| **Iterations** | 3 | 3 | 251 | 769 | 1726 | 2196 | 7147 | 10000 | 10000 | 10000 |
| **Time** | 1.72 | 1.8 | 5.85 | 14.68 | 25.97 | 32.28 | 98.26 | 140.129 | 140.74 | 141.91 |

From the above table, it is seen that the accuracy increases with decrease in threshold upto threshold = 1e-06, that is 0.000001, and then becomes constant. After that, the overhead (like time and number of iterations) starts to increase exponentially without any significant performance boost. Hence for logistic regression, the ***optimal threshold value should be 0.000001*** to obtain best results.

The following graph gives the train and test accuracy (using the optimal beta value) as a function of number of gradient descent iterations.



**Experiment 3:**

**Picking 8 features at random:** The gradient descent algorithm is executed two times, the 1st time for reduced model with 8 features selected at random, and the 2nd time for full model, keeping all other parameters constant.

*Linear Regression:*

The features selected arbitrarily (with no preference) are MWG, NWG, MDIMC, NDIMC, MDIMA, KWI, VWM and STRM.

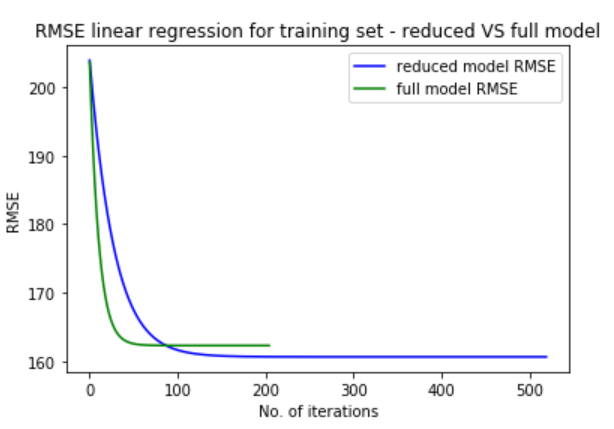
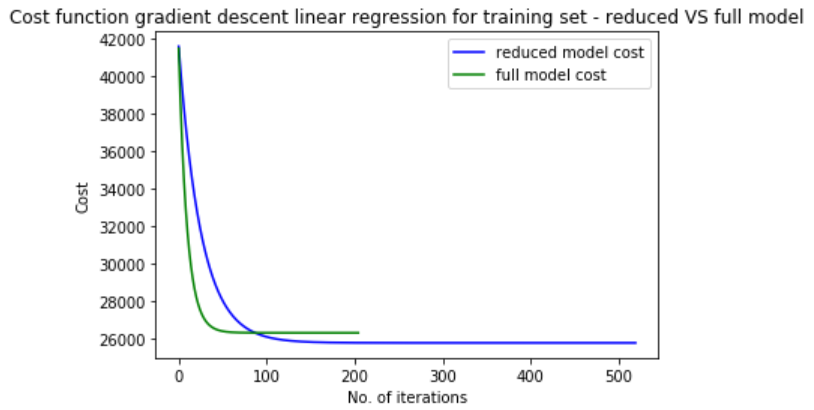
The reduced model regression equation becomes:

Run\_time = β0 + β1\*mwg + β2\*nwg + β3\*mdimc + β4\* ndimc + β5\* mdima + β6\* kwi + β7\* vwm + β8\*(strm\* mdima)

The optimal beta values for reduced model are:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 |
| 40.47 | 40.50 | 40.82 | 40.52 | 41.32 | 40.45 | 40.95 | 40.75 | 41.27 |

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Reduced Model** | **Full model** |
| Min. cost for training set | 25798.43 | 26332.56 |
| RMSE for training set | 160.619 | 162.27 |
| Estimated test cost | 25709.07 | 26268.56 |
| Estimated test RMSE | 160.34 | 162.07 |



From the above table and plots, it is seen that the reduced model performs better than full model. Choosing these 8 features and dropping other features improves the model.

*Logistic Regression:*

The features selected arbitrarily (with no preference) are NWG, KWG, NDIMC, MDIMA, NDIMB, VWM, STRM and SB.

The reduced model regression equation becomes:

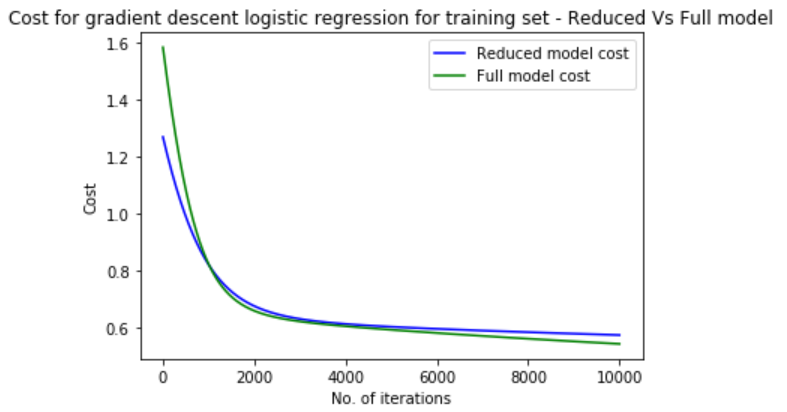
Run\_time (0/1) = β0 + β1\*nwg + β2\*kwg + β3\*ndimc + β4\*mdima + β5\*ndimb+ β6\*vwm + β7\*(strm\* mdima) +

+ β8\*(sb\*nwg)

The optimal beta values for reduced model are:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 |
| -0.856 | 0.212 | -0.249 | -0.378 | 0.3929 | -0.2609 | 0.5411 | -0.1340 | 0.055 |

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Reduced Model** | **Full model** |
| Min. cost for training set | 0.5719 | 0.5410 |
| Accuracy of training set | 0.707 | 0.7259 |
| Estimated test cost | 0.5725 | 0.5407 |
| Accuracy of test set | 0.7066 | 0.7247 |



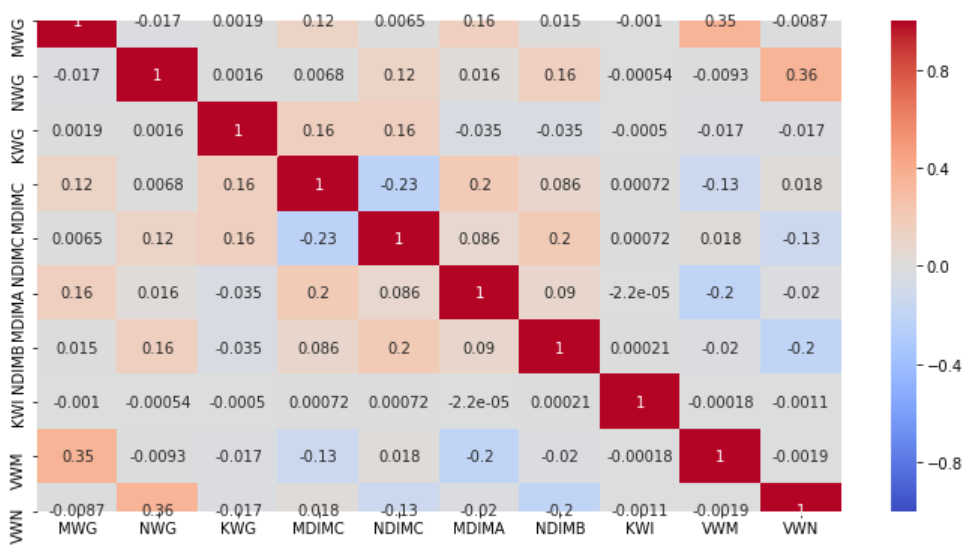
From the above table and graph, it is seen that the Full model slightly performs better than reduced model. Choosing these 8 features is not improving the full model.

**Experiment 4:**

**Picking 8 best features:** Feature selection isdone to pick 8 best features. The model is rerun using the best features and metrics are re-evaluated.

1. **Plotting correlation heatmap**

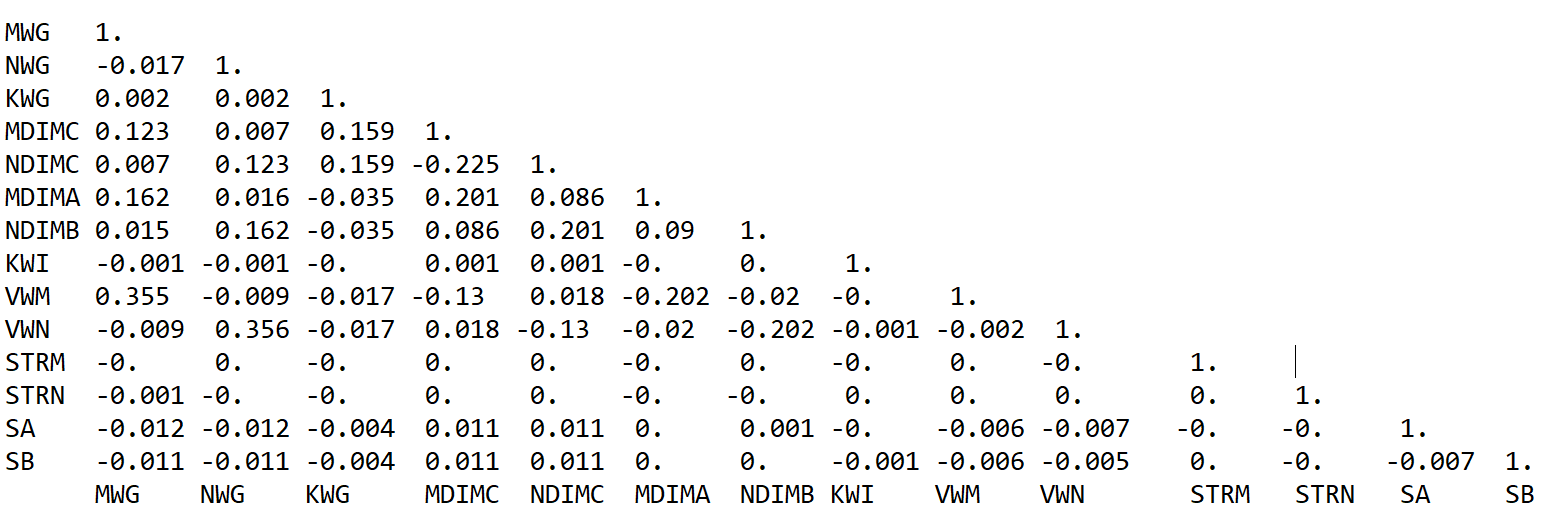
Heatmap is generated to find out the correlations among the predictor variables. No 2 variables have significant positive or negative correlation. Hence there is no problem of multicollinearity.



From the map, we find that NWG, VWN, VWM, MWG, MDIMC, NDIMC, MDIMA, NDIMB have better correlations than other predictors.

1. **Calculating Pearson’s correlation coefficient**

Pearson’s correlation coefficient is calculated and the predictors having maximum value for the coefficient are considered for the best model.



It is seen variables NWG, VWN, MWG, VWM, NDIMC, NDIMB, MDIMC, MDIMA have high Pearson’s correlation coefficient. Hence, they are chosen for the best model.

The gradient descent algorithm is executed two times, the 1st time for reduced model with the above 8 features, and the 2nd time for full model, keeping all other parameters constant.

*Linear Regression:*

The features selected are MWG, NWG, MDIMC, NDIMC, MDIMA, NDIMB, VWM and VWN.

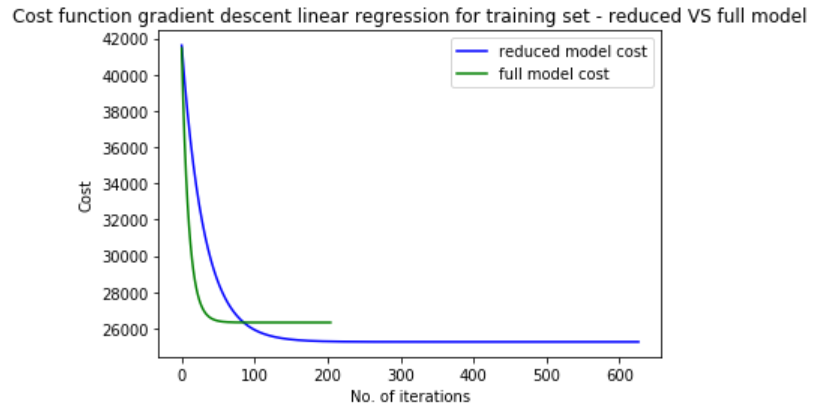
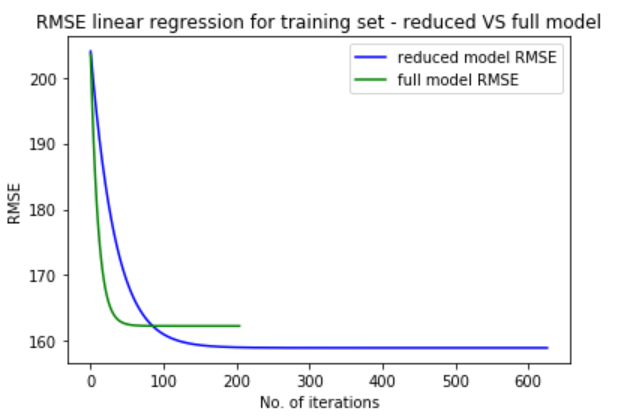
The reduced model regression equation becomes:

Run\_time = β0 + β1\*mwg + β2\*nwg + β3\*mdimc + β4\* ndimc + β5\* mdima + β6\*ndimb + β7\* vwm + β8\* vwn

The optimal beta values for reduced model are:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 |
| 45.378 | 45.41 | 45.73 | 45.43 | 46.22 | 45.36 | 45.86 | 45.65 | 46.17 |

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Reduced Model** | **Full model** |
| Min. cost for training set | 25263.719 | 26332.56 |
| RMSE for training set | 158.94 | 162.27 |
| Estimated test cost | 25166.536 | 26268.56 |
| Estimated test RMSE | 158.639 | 162.07 |

From the above table and plots, it is seen that the reduced model performs better than full model. Choosing these 8 features and dropping other features improves the model.

*Logistic Regression:*

The features selected are MWG, NWG, MDIMC, NDIMC, MDIMA, NDIMB, VWM and VWN.

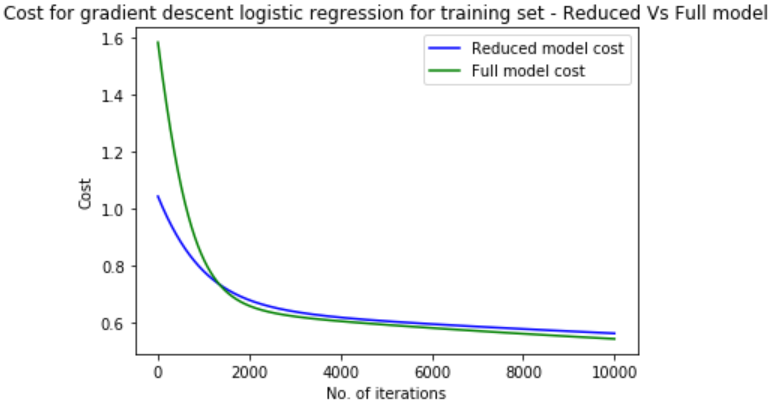
The reduced model regression equation becomes:

Run\_time (0/1) = β0 + β1\*mwg + β2\*nwg + β3\*mdimc + β4\* ndimc + β5\* mdima + β6\*ndimb + β7\* vwm + β8\* vwn

The optimal beta values for reduced model are:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| β0 | β1 | β2 | β3 | β4 | β5 | β6 | β7 | β8 |
| -1.046 | 0.177 | 0.2566 | -0.413 | 0.1944 | -0.308 | 0.0441 | 0.2917 | 0.7267 |

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Reduced Model** | **Full model** |
| Min. cost for training set | 0.5603 | 0.5410 |
| Accuracy of training set | 0.7195 | 0.7259 |
| Estimated test cost | 0.5613 | 0.5407 |
| Accuracy of test set | 0.7189 | 0.7247 |



From the above table and graph, it is seen that the Full model slightly performs better than reduced model. Choosing these 8 best features is not improving the full model.

**Discussion:**

From the model, we observe that NWG, VWN, MWG, VWM, NDIMC, NDIMB, MDIMC, MDIMA are the important parameters in determining the GPU run time. From the above linear and logistic models, it is found that the efficiency of the model is around 70%. The reduced Linear model better fits the data than the reduced logistic model.

* To get better results from model, domain knowledge about the hardware, manufacturing style, types of data processed, frequency of processing, etc, is required to predict the correct predictors in the model.
* Influence of other variables outside the given dataset need to be analyzed, and their significance need to be estimated. This will remove the endogeneity problem, leading to unbiasedness and obtaining accurate results.
* Different models like log-log, log-linear, etc should be run in order to capture the best fit equation. Since the nature of the true model is not known, and most models are heteroskedastic in nature, more experiments need to be performed with different types of models.
* Care should be taken to cleanse the data, scale and remove the outliers to remove their influence on the regression model. Else the model can be skewed.
* When multiple models are run, we can evaluate the best model using AIC, BIC, etc and compare their values.