BUAN 6341: Applied Machine Learning – Assignment 1

G.M.Sampath (MSG170330)

Data set description

- In this project we Implement a linear and logistic regression model on the dataset downloaded from UCI Machine Learning repository. This regression is to predict the GPU run time.
- Dataset consists of 241600 observations on 14 variables.
- Best model is identified by trail and error method by tuning the parameters like learning rate, threshold, iterations.
- For linear regression dependent variable is 'avg' which is average of four run times.
- For logistic regression the dependent variable 'avg' was converted into a binary variable using median as the threshold.
- Data split is done by 70:30 ratio for training and testing data set.
- No missing values found in the data set.

Correlation Plot

As part of preprocessing data we draw correlation plot. It lets us define the degree to which two variables are interrelated. It gives the correlation between independent variables. If two variables are highly correlated, the errors cause uncertainty. So when two variables are highly correlated, we pick either of them to create the model.

	MWG	NWG	KWG	MDIMC	NDIMC	MDIMA	NDIMB	KWI	VWM	VWN
MWG	1	0.0006	0.0093	0.11	-0.0086	0.16	0.015	0	0.35	-0.0008
NWG	0.0006	1	0.0093	-0.0086	0.11	0.015	0.16	0	-0.0008	0.35
KWG	0.0093	0.0093	1	0.15	0.15	-0.035	-0.035	-0	-0.012	-0.012
MDIMC	0.11	-0.0086	0.15	1	-0.21	0.2	0.085	-0	-0.13	0.011
NDIMC	-0.0086	0.11	0.15	-0.21	1	0.085	0.2	-0	0.011	-0.13
MDIMA	0.16	0.015	-0.035	0.2	0.085	1	0.088	-0	-0.2	-0.019
NDIMB	0.015	0.16	-0.035	0.085	0.2	0.088	1	-0	-0.019	-0.2
KWI	0	0	-0	-0	-0	-0	-0	1	-0	-0
VWM	0.35	-0.0008	-0.012	-0.13	0.011	-0.2	-0.019	-0	1	0.0012
VWN	-0.0008	0.35	-0.012	0.011	-0.13	-0.019	-0.2	-0	0.0012	1

Task 1

Data is Downloaded and partitioned into train and test set by 70 and 30 percentage.

Task 2

Linear regression model to model the average GPU run time is

avg = β0 + β1 * MWG + β2 * NWG + β3 * KWG + β4 * MDIMC + β5 * NDIMC + β6 * MDIMA + β7 * NDIMB + β8 * KWI + β9 * VWM + β10 * VWN + β11 * STRM + β12 * STRN + β13 * SA + β14 * SB.

Task 3

On running gradient descent algorithm with batch update rule our initial parameters will be

```
In [55]: gradientDescent(X_train,y_train,theta,iters,alpha)
```

```
array([[ 2.11335815e+02,  1.28600626e+02,  1.19413657e+02,
         3.10379210e+01, -1.13295748e+02, -1.11287442e+02,
         8.25584262e+00,  8.35313861e+00,  1.15627516e+01,
         9.40184755e+00,  4.88811497e+00, -4.26126207e+00,
         2.43452510e-02,  1.87262540e+01,  2.33421089e+01]])
```

Task 4

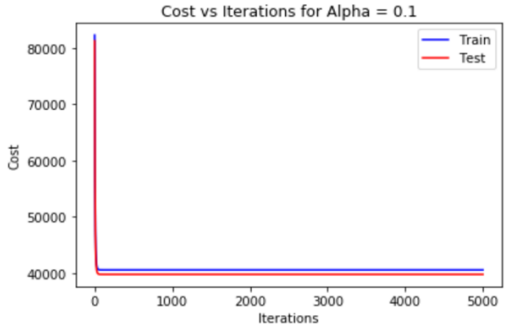
Converted the problem into a binary classification problem by calculating the median of the target variable. If the values of the target variable are less than median than classify them as 0 else 1.Accuracy metrics for train and test data sets

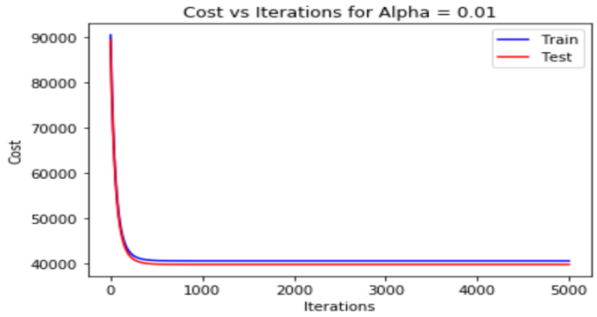
```
Confusion Matrix :
[[18534 17672]
 [ 3950 32324]]
Accuracy Score : 0.7016832229580574
Report :
              precision    recall  f1-score   support

         0.0       0.82      0.51      0.63     36206
         1.0       0.65      0.89      0.75     36274

    accuracy                           0.70     72480
   macro avg       0.74      0.70      0.69     72480
weighted avg       0.74      0.70      0.69     72480
```

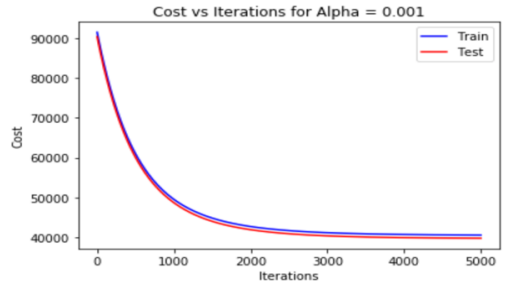
EXPERIMENT 1:

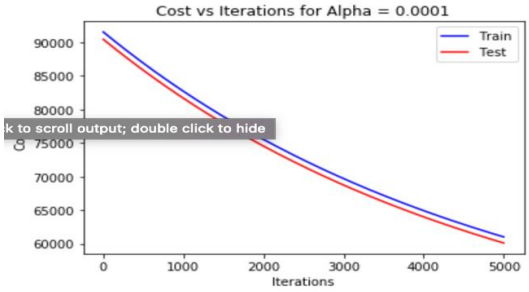
Experiment with various parameters for linear and logistic regression (e.g. learning rate ∝)and report on your findings as how the error/accuracy varies for train and test sets with varying these parameters. Plot the results. Report the best values of the parameters.

Linear Regression: We had chosen Alpha=0.1,0.01,0.001,0.0001

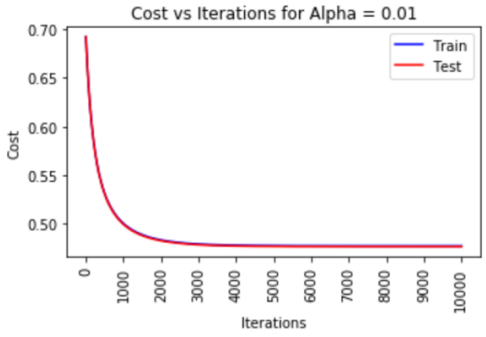


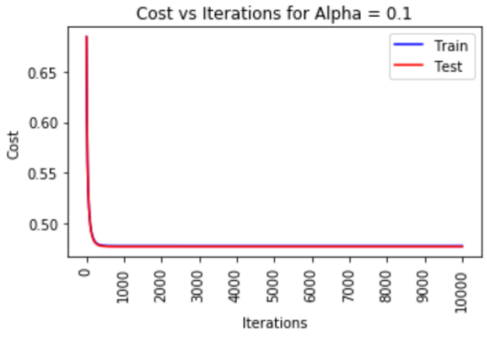


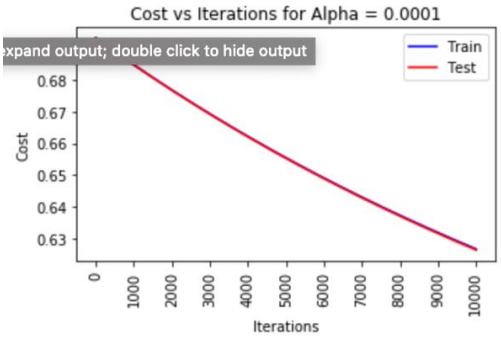


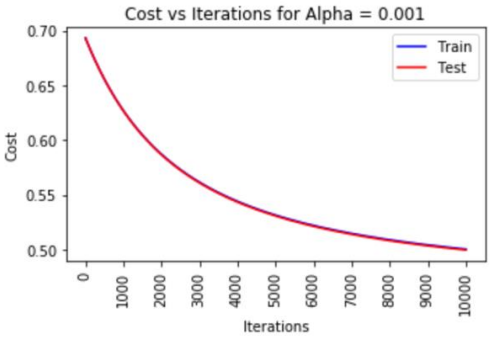


Logistic Regression:



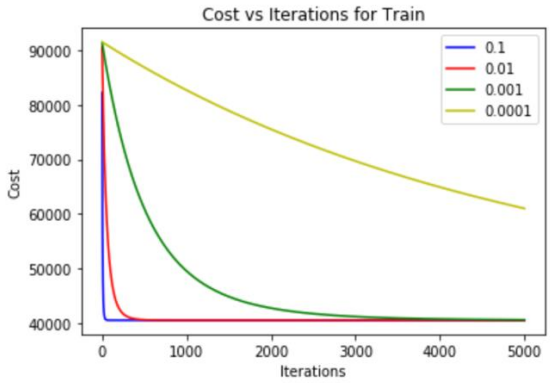


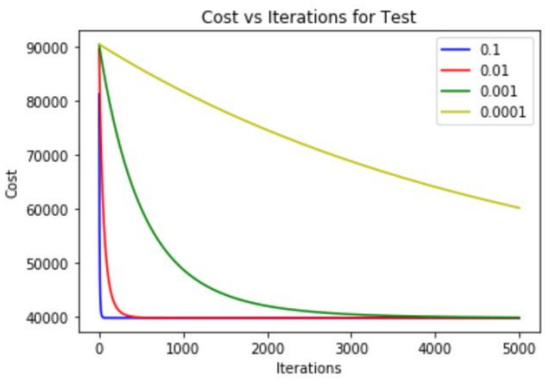




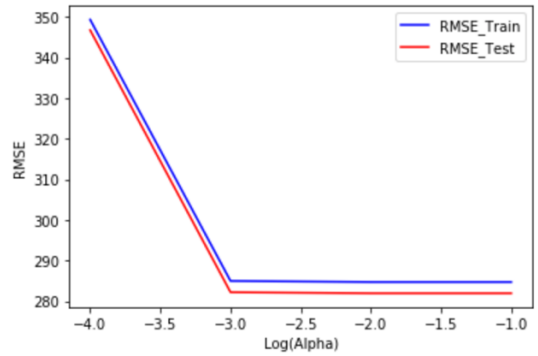
Cost decreases with increase in iterations and for given alpha, after certain iterations there is negligible change in the cost which implies the gradient descent algorithm has reached convergence. Optimal values occurs at 0.1 and 0.01.

Linear Regression:

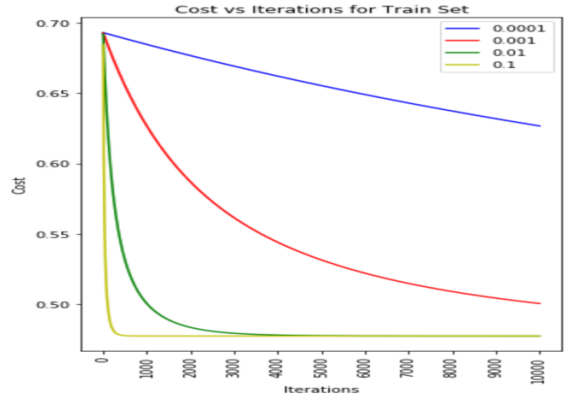


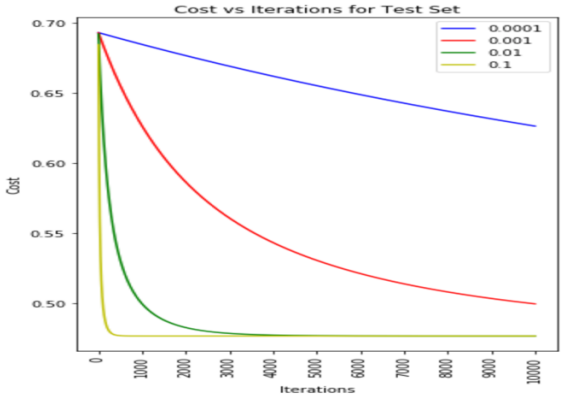


There is a decrease in the cost for both training and test sets. The optional value of the alpha is the one, where the cost converges with a smaller number of iterations. From the above we can say that the optimal value of alpha can be 0.01 or 0.1 which can be confirmed by alpha vs RMSE plot.



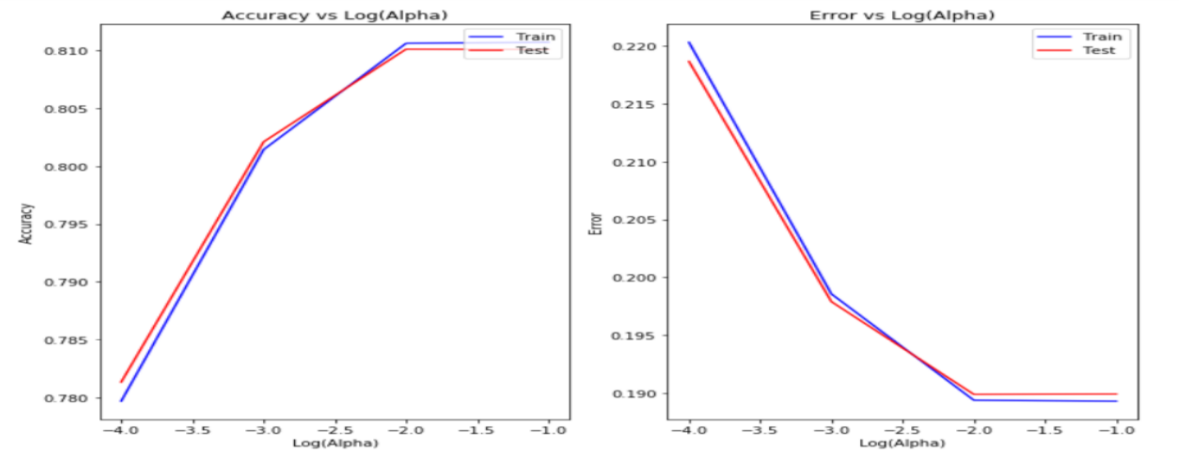
Logistic Regression:





Decrease in cost is observed with the increase in iterations.Optional value is 1,optimal value of alpha is 1. Optimal value of alpha can be 0.01 or 0.1 which can be confirmed by alpha vs RMSE plot

Logistic Regression:



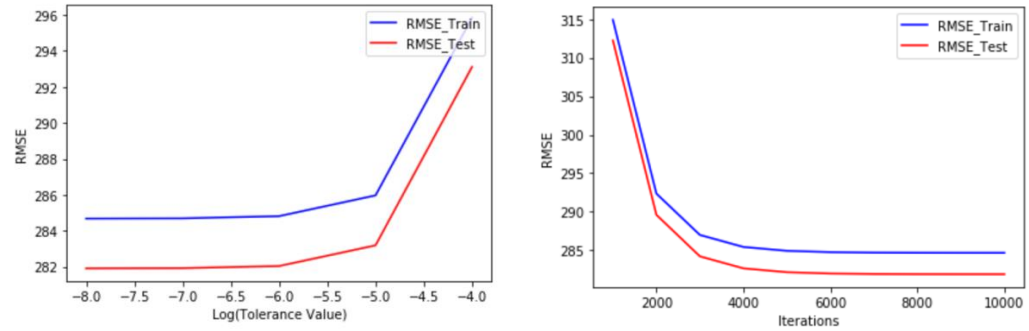
Decrease in cost is observed for both the train and test data's w.r.t iterations. When cost converges with minimal number of iterations optimal learning rate occurs. Optimal learning rate to choose is 0.1.

The best learning rate is observed by RMSE vs Alpha plot. High RMSE is observed at 10^-3 and similar for other learning rates as the gradient descent is converged.
Parameters:

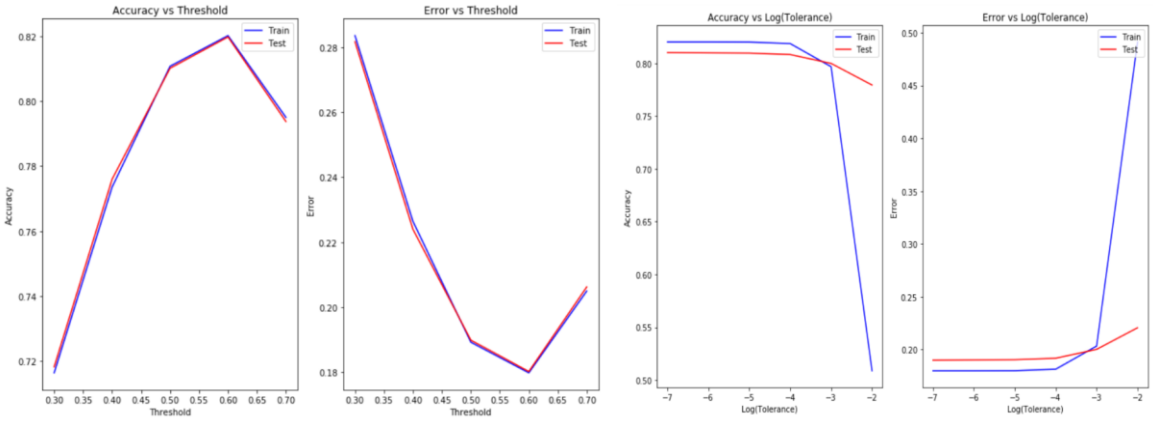
```
array([[ 2.11335815e+02,  1.28600626e+02,  1.19413657e+02,
         3.10379210e+01, -1.13295748e+02, -1.11287442e+02,
         8.25584262e+00,  8.35313861e+00,  1.15627516e+01,
         9.40184755e+00,  4.88811497e+00, -4.26126207e+00,
         2.43452510e-02,  1.87262540e+01,  2.33421089e+01]])
```

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.
Linear: Different values of threshold to run gradient descent are 10^-8, 10^-7, 10^-6, 10^-5, 10^-4.



RMSE is high for high tolerance values, as the algorithm converges for large change in cost RMSE is similar after tolerance value of 10^-5.Optimal value for learning rate is 10^-5.After 4000 iterations RMSE is same for both train and test data.
Logistic: Different values of threshold to run gradient descent are 0.3,0.4,0.5,0.6, 0.7.



From the above graphs we can infer that, the accuracy increases with increase in the threshold upto certain point i.e. 0.60 and then decreases similarly the error decreases at 0.60 threshold. Hence 0.60, is the optimal value of threshold.

EXPERIMENT 3 & 4:

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.

Linear:

Random sample had been picked from the given variables using Rand function. Model is retrained with the random samples. Calculated RMSE for the random variables is

```
print(rmse_train8)
print(rmse_test8)
print(tol_rmse_train[3])
print(tol_rmse_test[3])
```

```
339.43777835154344
332.3165897886614
285.6560468516569
279.89802806727636
```

Based on the beta values top 8 predictors are chosen and all the functions are performed with those values. Calculated RMSE for best picked values is

```
print(rmse_test_best)
print(rmse_train_best)
```

```
280.28012020177977
286.15684947519884
```

We can see that the RMSE values for the chosen predictors provides better results.

Logistic:

Random sample had been picked from the given variables using Rand function. Model is retrained with the random samples. Calculated Accuracy for the random variables is

```
print(acc_train8)
print(acc_test8)
```

```
0.8224061810154525
0.8225579470198675
```

Based on the beta values top 8 predictors are chosen and all the functions are performed with those values. Calculated Accuracy for best picked values is

```
print(acc_train_best)
print(acc_test_best)
```

```
0.8172047461368653
0.8195364238410596
```

We can see that the Accuracy for the chosen predictors provides better results.

Discussion:

The other step which we can take for modelling are using stepwise, backward, forward regression techniques for feature selection instead of randomly choosing them.