Logistic Regression

-Rahul Mehndiratta 2017A7PS1479H
-BVS Ruthvik2019A7PS0017H
-R Vedang. 2019A7PS0150H

Problem Statement:

1. In this assignment, you will implement logistic regression from scratch for binary classification. To train the model, you will use Gradient Descent and Stocastic Gradient Descent with appropriate learning rates. Plot the loss and accuracy for your model every 50 iterations to visualize the training better. You are expected to create 10 independent random 70:30 splits on the given data, train the model and report the average loss and accuracy over all those 10 splits.
2. Try to vectorize your code as much as possible to make your computations faster and efficient. Do not hard code any parts of the implementation unless it is absolutely necessary.

Introduction :

We trained the model using Gradient Descent and Stochastic Gradient Descent with different learning rates. The loss and accuracy for the model has been plotted every 50 iterations to visualize the training better. 10 independent random 70:30 splits on the given data were created , the model was trained and the average loss and accuracy over all those 10 splits was calculated.

Algorithm :

Predict Function : 1 if i>0.5 else 0
The Loss function is given as : $L(x) = - t * \log(y) - (1-t) * (\log(1-y))$
The gradient of loss function : $\sum (Yn - Tn). Xn$
First, we initialize the weight vector randomly and in each iteration update the weight vector by moving alpha(Learning rate) steps towards the gradient of the loss function.

This will continue till the difference between the weight vectors in consecutive iterations is less than a particular threshold.

Data:

Features : 4 Entries : 1372
Class : 0/1
Standardised the data and 10 independent random 70:30 splits

Observations :

The most important dataset in the feature was found by scaling each attribute in the dataset and then running a model over the dataset for a few iterations and the average of the weights were obtained. The weight/coefficient with the highest absolute value was taken as the most important one.
For GD 2nd attribute was found to be most important.

```
=====================================Average values :=====================================

.....................................Gradient Decent.....................................

GD weights:  [-10.92488133 -12.27743461 -11.22176209  -0.21290279]
GD bias:  -5.010051948409773
Loss : 21.732220213612752
```

For SGD 1st attribute was found to be most important.

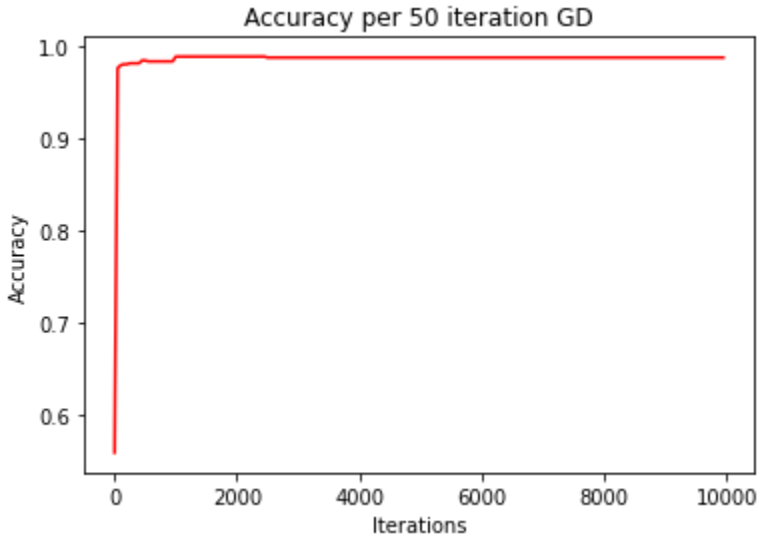
```
.............................Stochastic Gradient Decent.............................

SGD weights:  [-1.51998227 -0.90178568 -0.41564012 -0.0189702 ]
SGD bias:  -0.2405082907264002
Loss : 0.18661987826743448
```

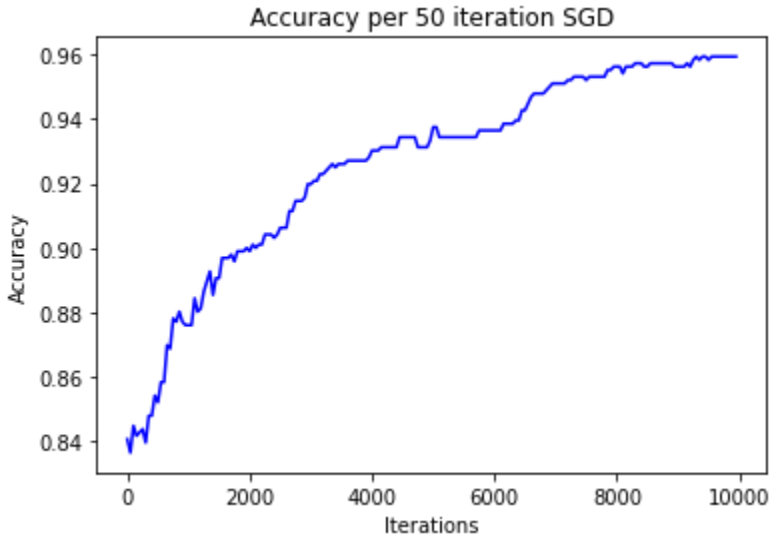
Plots for 3 different Learning Rates:

----------Learning Rate : 0.002 ----------

Final GD weights: [-12.53978716 -14.35998534 -13.05680357 -0.53157607]
Final GD bias: -5.987052742842129
GD accuracy: 0.9885416666666667

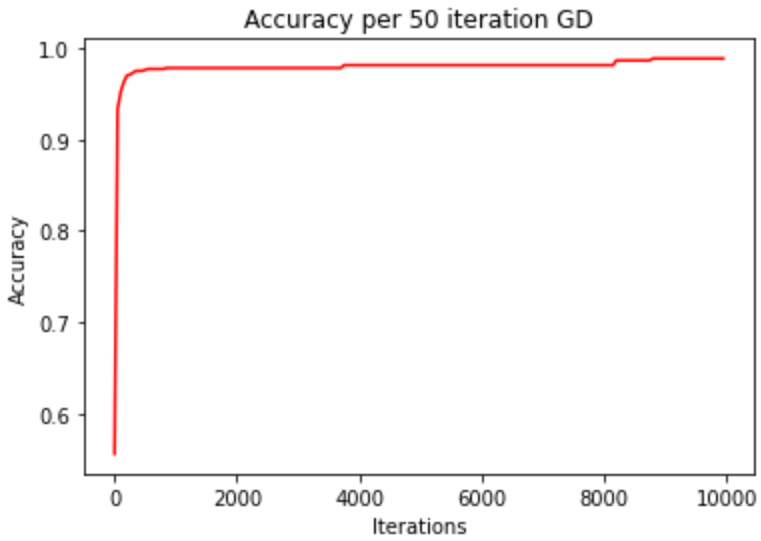


Final SGD weights: [-2.06576115 -1.34847752 -0.95808907 0.04193729]
Final SGD bias: -0.34161050455333003
SGD accuracy: 0.959375

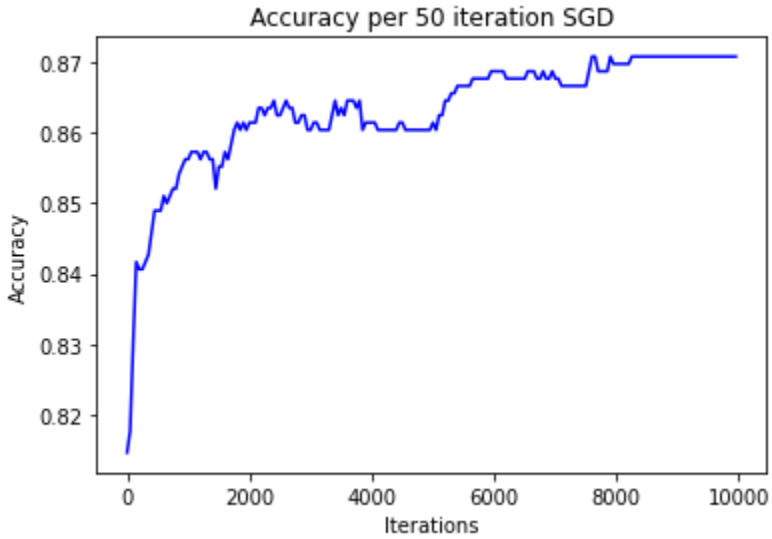


----------Learning Rate : 0.0002 ----------

Final GD weights: [-7.05341301 -7.72509585 -6.98070543 0.02715194]
Final GD bias: -2.857036477173407
GD accuracy: 0.9885416666666667

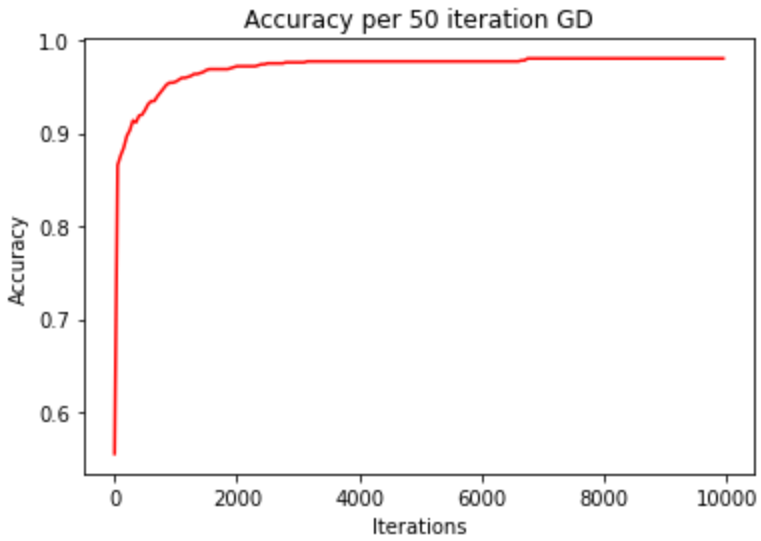


Final SGD weights: [-0.54519101 -0.31135257 0.02473814 -0.02302347]
Final SGD bias: -0.10392806812544982
SGD accuracy: 0.8708333333333333



----------Learning Rate : 2e-05 ----------

Final GD weights: [-4.07292764 -3.85381162 -3.59503798 0.28604037]
Final GD bias: -0.9498329148474322
GD accuracy: 0.9802083333333333



Final SGD weights: [-0.06839715 -0.04270911 0.01152766 -0.00147009]
Final SGD bias: -0.00988455951828226
SGD accuracy: 0.8541666666666666