CS584 HOMEWORK 2
NAME : Preeti Bhattacharya (G01302375)
USERNAME IN MINER : Preeti, **ACCURACY SCORE**: 0.91, **RANK**: 23 (At time of Upload)

Introduction:
Logistic Regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable. Here it contains data coded as 1 (heart attack.) and -1 (no heart attack).
In this assignment I implemented a basic Logistic regression using gradient decent to classify a dataset of 152 heart attack cases data that was provided.

Approach:
I spent the initial days trying to learn about different libraries, what they do and how they are implemented, as I have very basic knowledge about data mining.

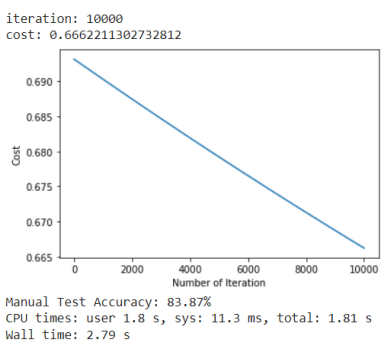
1) **Data framing**:-I replaced -1 (no heart attack) with 0 and renamed heartdisease::category|-1|1 with “final” for easy understanding.
2) **Data pre-processing**:-Then I normalized data as part of data preparation to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values and to avoid overfitting.
3) **Data splitting**:- Then I divided the dataset into two sections, with 80 percent (121 instances) used for training and 20 percent (31 instances) used for testing.
4) **Training Model**:-After splitting I have initialized weight and bias with zero. Then I calculated Sigmoid function.
5) Then I applied a condition if gradient is less than 10^{-3} it will terminate the algorithm.
6) After that I calculated cross-entropy error and gradient decent function using

$$\theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

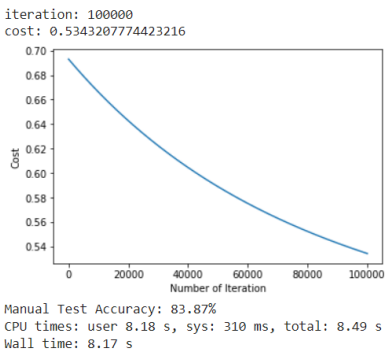
7) I used these values to train my model using Logistic regression. Where I gave learning rate as 10^{-5} and gave different iterations to find the cost and time of the model.
8) After training the model I found the classification error and did generalization. Both are explained in detail below with values.
9) I have also used sklearn LogisticRegression model for comparing my implemented model to show that my trained model is better when compared with it.

Cross entropy loss function is used as an optimization function to estimate parameters for logistic regression models. We have used gradient descent algorithm with cross entropy loss function to estimate the model parameters and we got the below table.

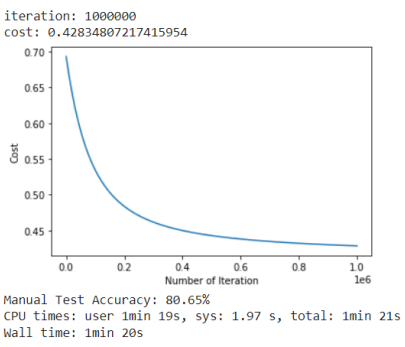
	Eta	Iteration	Cost of Cross-Entropy	Manual Accuracy	Test Accuracy	Miner Accuracy
1	10^{-5}	10,000	0.6662	83.87%	80.65%	91%
2	10^{-5}	100,000	0.5343	83.87%	80.65%	91%
3	10^{-5}	1,000,000	0.4283	80.65%	80.65%	91%



(1)



(2)



(3)

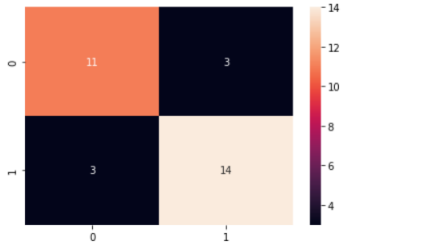
Classification problems are usually binary identification determining if an observation is, or is not, a certain condition. Out of any classification model there are four types of results. I would like to introduce the meaning of TP,FP,TN and FN. A true positive (TP) is a positive outcome predicted by the model correctly while a false positive (FP(Type I)) is a positive outcome predicted by the model incorrectly. A true negative (TN) is a negative outcome predicted by the model correctly while a false negative (FN(Type II)) is a negative outcome predicted by the model incorrectly. So here while training the model we are getting the below classification error for 1,000,000 iterations.

```
              precision    recall  f1-score   support

           0       0.79      0.79      0.79        14
           1       0.82      0.82      0.82        17

    accuracy                           0.81        31
   macro avg       0.80      0.80      0.80        31
weighted avg       0.81      0.81      0.81        31
```

```
[[11  3]
 [ 3 14]]
<matplotlib.axes._subplots.AxesSubplot at 0x7fc6a07bc050>
```



Iteration	Classification Error on Model	Classification Error on Miner	Time taken to train the model
10,000	(1-0.8387) = 0.1613	(1-0.91) = 0.09	1min 21s
100,000	(1-0.8387) = 0.1613	(1-0.91) = 0.09	8.49 s
1,000,000	(1-0.8065) = 0.1935	(1-0.91) = 0.09	1.81 s

Generalization refers to model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model. So, I have trained my model on 1644871288_9762487_cleveland-train.csv data, where I got an accuracy of 83.87% and then I used 1644871288_9775174_cleveland-test.csv data which was unseen and got an accuracy of 91%. The model has adapted good generalization as the accuracy on unseen data is greater than training data. The difference between training and test set classification errors are 7.13% for 10,000 and 100,000 iterations and 10.35% for 1,000,000 iterations.

CONCLUSION WITH COMPARISION AND FUTURE WORK:

The final result of this assignment is with 91% accuracy. As the dataset was very large and with a large number of features there is still more room for improvement in terms of feature extraction. I have standardized the data with sklearn StandardScaler().fit_transform() method which has given me better classification rate compared to Min_Max normalization. The Min_Max normalized data model was giving accuracy of 74%.

There are other methods that can be investigated such as KNN, SVM, Naïve Bayes Decision Tree, Random Forrest other than Logistic regression. The advantage of the Logistic Regression is that it does not need too much computational resources and it is highly interpretable. So, it is easy and sufficient to apply Logistic Regression. However, the limitation of Logistic Regression is that it assumes linearity between the features of the dataset. In the real world, the data is rarely separable, neither as our dataset. That is why we cannot reach a very high accuracy. Our models work fine but we can look for other approaches like KNN or Random Forest which can give better accuracy.

References:-

https://medium.com/@cdabakoglu/heart-disease-logistic-regression-machine-learning-d0ebf08e55c0
https://medium.com/geekculture/logistic-regression-implementation-from-scratch-in-python-f9d6cd4a0747
https://towardsdatascience.com/heart-disease-uci-diagnosis-prediction-b1943ee835a7