# ASSIGNMENT 2 – REPORT

## Loss Function:

This function is used to estimate the error in classifying the data using sigmoid function with respect to that of the skylearn's logistic regression function. This is also called error function.

We know that,

y_hat = $\sigma(z)$, where z= $\Sigma\, w_i x_i$ ; $x_i = coefficient\ of\ the\ point\ or\ feature\ value; w_i = weight\ factor$

Also, Maximum likelihood function is the product of probabilities of data points being correctly classified. This function can be used to compare models only when the volume of the data set is small. When considering larger data sets the value becomes comparatively inconvenient and hence this method can't be used to decide the best fit model to classify data. To avoid this problem, negative of logarithm is applied to the maximum likelihood function which returns the summation of negative logarithm of probabilities of all the data points. This is called **cross-entropy.** For a data set of 'n' points classified as 0|1, average log loss/ binary cross entropy is given as follows:

$$L = \frac{1}{n} \sum_{i=1}^{n} - y_i \log (y\_hat_i) - (1 - y_i) \log (y\_hat_i)$$

$$L = -\frac{1}{n} [y \log(y\_hat) + (1-y) \log(1-y\_hat)];\ y\_hat = \sigma(wx)$$

This error function does not have a closed form solution. So, the weight vector 'w' is calculated such that the error function has an optimized/minimal value. Hence, gradient descent is applied on the loss function to calculate the logistic regression of a specific dataset.

## Gradient Descent:

The dataset that is provided is of size (152,14). The last column is the 'y'. The weight vector for this dataset will have 14 elements (13 coefficients and 1 intercept). The predicted output 'y_hat' will have a size of (152,1).

We know that,

$$\frac{dL}{dW} = -\frac{1}{n} (y - y\_hat)\ X$$

Updating the co-efficients:

$$W = W - \eta \frac{dL}{dW}$$

$$W = W + \eta \frac{1}{n} (y - y\_hat)\ X$$

### Analysis on Training data:

Apply the loop on training data for different no. of iterations and update the weight vector with learning rate of 10^-5.

| Iterations | Time Taken |
|---|---|
| 10000 | 0.29973673820495605 |
| 100000 | 3.130495309829712 |
| 1000000 | 23.955679655075073 |

Here, we can see that as the no. of iterations increase the time taken increases keeping the eta /learning rate constant.

Cross entropy error before updating the weight vector (i.e., considering zero vector) on training data =

0.6931471805599453

Cross entropy error after updating the weight vector = 0.5847145522443018

Accuracy = 0.8157894736842105
Cross validation error/classification error = 0.18421052631578946

**Effect of learning rate:** Changing the learning rate and keeping the number of iterations constant (1 million) for training data

| Eta | Accuracy | Classification error | Cross-entropy error |
|---|---|---|---|
| 0.00001 | 0.8157894736842105 | 0.18421052631578946 | 0.43535260915252544 |
| 0.0001 | 0.8289473684210527 | 0.17105263157894737 | 0.4191702834417414 |
| 0.001 | 0.8355263157894737 | 0.16447368421052633 | 1.3588274758118926 |
| 0.01 | 0.6776315789473685 | 0.3223684210526316 | Nan |

We can see only a little variation in accuracy when eta value is altered between 10^-5 to 10^-3. But when eta value is further increased, there's a significant drop in accuracy from 83% to 67%. As the eta value becomes large the accuracy decreases and classification error increases.

**Iterations vs log loss/cross- entropy error:** After updating the weight vector log loss is calculated for different no. of iterations on training data. The results are as follows:

| Iterations | Cross-entropy error (training dataset) |
|---|---|
| 1000 | 0.616818027996214 |
| 10000 | 0.5847145522443018 |
| 100000 | 0.49370175927710563 |
| 1000000 | 0.43535260915252544 |

As the number of iterations increase the log loss/cross-entropy error on the training dataset decreases.

**Comparing with standard function:**

Using the "LogisticRegressionCV()" function from the skylearn library on the training-set.

Accuracy using inbuilt function = 0.8421052631578947

Accuracy using the gradient descent = 0.8157894736842105

**Normalizing the data and changing eta:**

| Eta | Time Taken | No. of iterations before loop termination | Accuracy |
|---|---|---|---|
| 0.000001 | 27.83063793182373 | 1000000 | 0.5657894736842105 |
| 0.000003 | 27.75358486175537 | 1000000 | 0.6710526315789473 |
| 0.000005 | 27.197702646255493 | 1000000 | 0.7039473684210527 |
| 0.000007 | 27.26084041595459 | 1000000 | 0.6973684210526315 |
| 0.000009 | 27.49708080291748 | 1000000 | 0.6907894736842105 |

Miner Username: ds18; Rank: 15; Accuracy score: 0.89; language used: python; Software requirement: google collab