**INTRODUCTION:**

In this component,we tried to understand and develop, optimization techniques for logistic regression models in the form of **Gradient descent using L1&L2 regularizations**.Using the given dataset.

**Methodology:**

We have developed a logistic regression model using gradient descent without any regularized terms.In this algorithm weights of the equation to be optimized are updated after each and every iteration.

We choose the sigmoid function as an activation function which is to be optimized.

ypred= 1/1+ e-wx+b

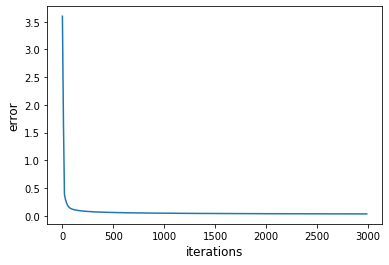
Error function of this is determined to be

Cost = -1/m[ y(i)log( ypred(x(i)))+(1-y(i))log(1-ypred(x(i)))]

We have implemented a variable learning rate starting with 100 and we divided the learning rate by 1.5 whenever we found the error to be diverging.Weights obtained after the development of the model are [-31.45347132, -30.15514933 -32.73909891, 1.36191249] ,40.76843884066982 after 3000 iterations with initial weights taken to be [1,2,3,4] ,0 with no reason.The final error converges to 0.03401931

With Accuracy= 97.8102189781022

And F-score =0.9752066115702478



**Ridge Regression and Lasso Regression (regularisation):**

Here we tried building the same model that we built in gradient descent part but with but by adding a constraint term to the cost function.

**Ridge Regression**

In ridge regression the constraint term is /m x (sum of squares of weights).

As it is gradient descent we are required to calculate partial derivatives of cost function.But since we changed the whole cost function we once again calculate all the partial derivatives.We split the data into 80 percent training set and 20 percent test set.We first train the model and find training loss and then test the model with test data and find validation loss.

Error function of this determined to be

cost= -1/m[ y(i)log( ypred(x(i)))+(1-y(i))log(1-ypred(x(i)))] + /m wi2

here w consists of weights and lambda is a regularisation parameter.

We start the training with following initialisation

Learning rate :100

Number of Epochs : 3000

Choosing (lambda):

Obtained model is tested with training set and testing set both have a difference in errors so we choose such a lambda which reduces the difference between training error and testing error

**Results:**

Overall we have observed an increase in error by regularisation which is a trade off we have to have a good reliable model.

Weights obtained after the development of the model are [-17.9791106, -15.56075689, -16.48076287, 0.87508762] ,22.045764914586204 initial weights taken to be [1, 2, 3, 4], 0 with no reason.The final validation error comes out to be 0.118750. And regularisation constant is 0.125 which is taken randomly after testing with few values.

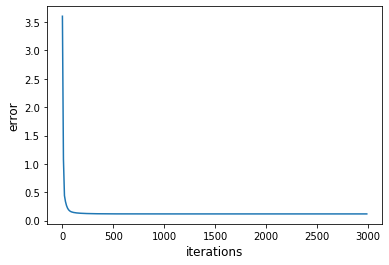
test set

97.8102189781022 0.9752066115702478

train set

98.08743169398907 0.978978978978979

Which represents Accuracy and F-score respectively



**Lasso Regression**

In Lasso regression the constraint term is /m x (sum of absolute values of weights).As is gradient descent we are required to calculate partial derivatives of cost function.But since we changed the whole cost function we once again calculate all the partial derivatives.We split the data into 80 percent training set and 20 percent test set.We first train the model and find training loss and then test the model with test data and find validation loss.

Error function of this determined to be

cost= -1/m[ y(i)log( ypred(x(i)))+(1-y(i))log(1-ypred(x(i)))] + /m |wi|

Here w consists of weights and lambda is a regularisation parameter.

We start the training with following initialisation

Learning rate :100

Number of Epochs : 3000

Choosing (lambda):

Obtained model is tested with training set and testing set both have a difference in errors so we choose such a lambda which reduces the difference between training error and testing error.

**Results:**

Overall we have observed an increase in error by regularisation which is a trade off we have to have a good reliable model.

Weights obtained after the development of the model are [-19.24324139, -13.16016656, -15.67773709, 3.00480571], 19.716849344792095 initial weights taken to be [1,2,3,4], 0 with no reason.The final validation error comes out to be 0.2169168688.And the regularisation constant w.

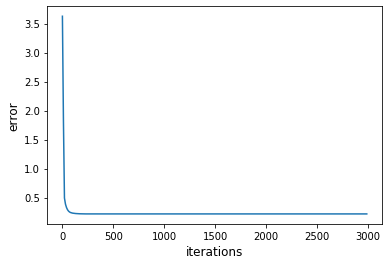
test set

97.44525547445255 0.97119341563786

train set

97.44990892531877 0.9719438877755511

Which represents Accuracy and F-score respectively



# tough starting with different distributions of weights for optimization we end up nearly at the same weights for same learning rate and same iterations

# If we train the model for longer periods weights are varying much from lesser no of iterations but accuracy and f score remains almost the same.this might be because less gradient convex curve

#We have scaled all the features (inputs) to range from 0-1.This is giving better results because sigmoid can differentiate values if x is in the vicinity of 0.