Introduction

Logistic Regression expands on linear regression as a classification algorithm which has only two possible outcomes, which are known as classes. It is a classification model that uses input variables to predict a categorical outcome variable that can take on one of a limited set of class values. Logistic regression uses the sigmoid function to weighted input values to generate a prediction of the data class.

Logistic regression is used when

* Output to be divided into 2 classes.
* Decision boundary can be used to split the data into binomial responses easily

Dataset

We are given a problem to predict whether a tumor is malignant or benign based on the dataset provided by WDBC. The Dataset consists of 569 rows of data specific to the characteristics of tumor like its radius, area, smoothness, etc., each having 32 attributes. Out of 32 features, 31 are independent variable and the column (say, result) with values {‘M’, ‘B’}, is a dependent variable, whose value is predicted by the rest 30 features. 1 feature i.e. ID is redundant as it does not affect the value of result.

Preprocessing

Logistic Regression is implemented using following processes:

* Input dataset is read through pandas:

The ‘pandas’ library is being used to read the dataset file into python code. The header is set as none, since the dataset has not been provided with headers. Following is the extracted dataset using pandas:



* Mapping Column 1:

The dataset has output tokens as {‘M’, ‘B’}, where M indicates class ‘Malignant’ and B indicates class ‘Benign’ Cancer. It is important to change this indicator as the whole dataset would be comprised of float values.

* Dataset classification:

The 1st column is considered redundant and is not used in Input nor Output. Data is classified into Training data (80% of the main dataset), Validation data (10% of the main dataset) and Test data (remaining 10%) using manual classification using numpy arrays. Here, the training data is used to train the logistic regression model. Weighed values are used in validation data to obtain optimum values of hyperparameters and test data is used to obtain the result of the trained model on real set of data.

* Normalization:

30 features in the dataset representing a different characteristic of tumor has a chance that one of the features may have a higher range e.g. between 500-5000 and some may have very low range e.g. between 0.001-1. This ill-affects the regression as the low-range values may become redundant. To avoid redundancy, normalization of each data value is done.

* Initializing w and b:

w is assigned with an array of random values and b is assigned with 0.

Algorithm

The algorithm is constructed as 3 parts:

* Gradient Descent:

For logistic regression, the hypothesis is sigmoid of ‘*wTX+b’*. We have to find values of *w* and *b* such that it minimizes the cost function. To overcome the issue of ‘*wTX+b’* ranging from 10 to 10000, Sigmoid function is used to create a threshold which basically gives the probability that *wTX+b* is very big or small. Where, the condition will be either 0 or 1.

The process of minimizing the gradient descent has 2 process,

* Forward Propagation: where the above mentioned *wTX+b* is proceeded by sigmoid to calculate the input for the log values in cross entropy loss used to calculate cost. 
* Back Propagation: Weights are approximately updated for efficient minimization of loss function. w and b are updated with its respective values for the next iteration which again carry forwards the same process to minimize the loss.



* Predict function:

This function is basically used to predict the values of the vector by assigning 1 if that particular value is greater than 0.5 otherwise 0.

* Calling the Logistical model:

The model is a combination of Gradient Descent and Predict function. After prediction vectors are obtained from the model, confusion matrix is used to calculate Accuracy, Precision and Recall.

The results are obtained as shown below.

Results



*Figure 1 Iteration number: 20000, learning rate: 0.5*

In Figure 1, the cost function value for each epoch while training with 0.5 learning rate is mapped. We can observe that; the cost function gradually reduces to a minimum value.



*Figure 2 Cost and output of Accuracy, Precision and Recall of Train, Validation and Test data*



*Figure 2 cost v/s epochs*

In Figure 2, the cost function with respect to multiple learning rate is mapped.

Learning rates used are: 0.3 for 25000 iterations, 0.1 for 30000 and 0.5 for 20000 iterations

We can observe that; the cost function gradually reduces to a minimum value.

Conclusion

Thus, Logistic Regression is implemented by having training data train with an accuracy of 91%, modified the hyperparameters to achieve 91% is validation data and predicted the output for test data with an accuracy of 91%.