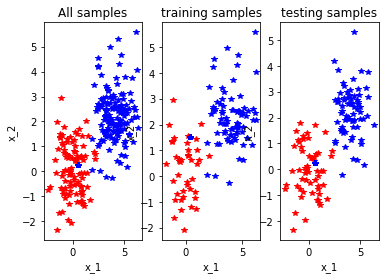
**Assignment 3 Report  
Machine Learning (CS-596)**

**Name: Dhaval Harish Sharma  
Red ID: 824654344**

**Question 1:  
Introduction:**This program makes a dataset containing two features as input and calculates the binary labels based on two feature vectors by using Logistic Regression as output. It uses two implementations of the logistic regression to train the model. It makes a confusion matrix to evaluate the model’s performance.

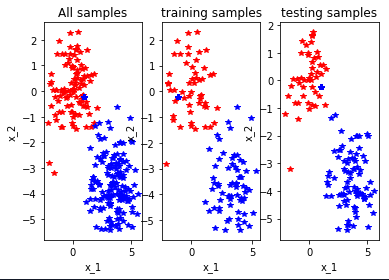
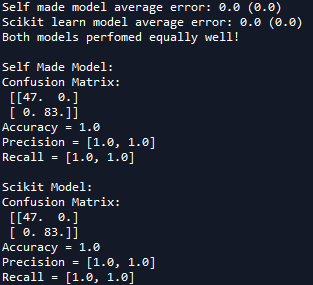
**Working of the program:**The program starts with importing the necessary libraries for its working. “getDataset.py” file contains logic for making the dataset. Numpy is a library which provides a powerful array with necessary mathematical computations. Matplotlib is used for plotting the results of the processing.

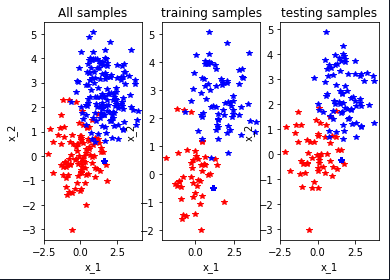
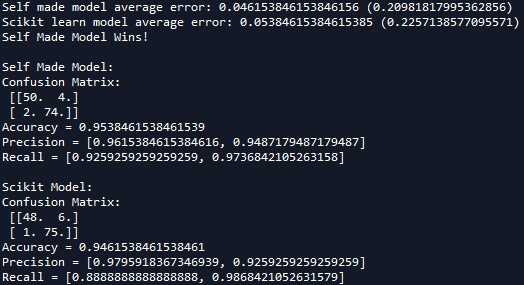
There is a function called func\_DisplayData which displays the data samples passed as parameters to it in the form of graph using matplotlib in the output. Then, we choose the value of training samples in the entire dataset. We create random indexes using np.random.choice which helps to split the data in to testing and training dataset. After splitting the dataset in to training and testing, we display all the datasets in the form of graph in the output. One of the sample output looks as follows,  


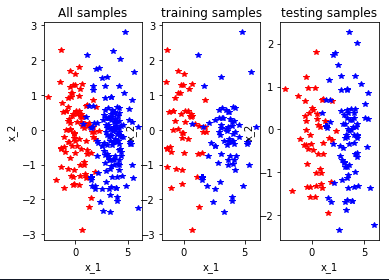
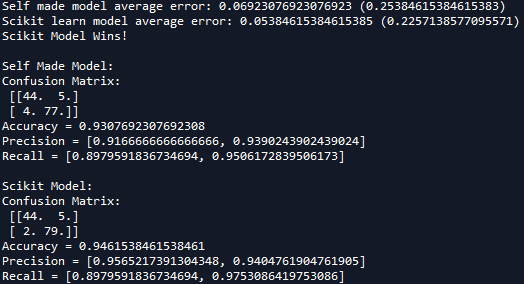
Then, we train the model on the training dataset using logistic regression. There are two implementations for training the model. The first one is the self made model which is the implementation of the logistic regression from scratch. The second one is by using the logistic regression from the scikit learn library. Initially, we initialize the values of theta, alpha and the max iterations. We append the feature vector with ones to make x0. Then, we start the loop for finding the value of theta and simultaneously finding array cost. The gradient descent minimizes the cost function by finding the direction of movement and the amount of movement. There is a sigmoid function which reduces the range of the output values between 0 and 1. The other implementation uses LogisticRegression function from the scikit learn library.

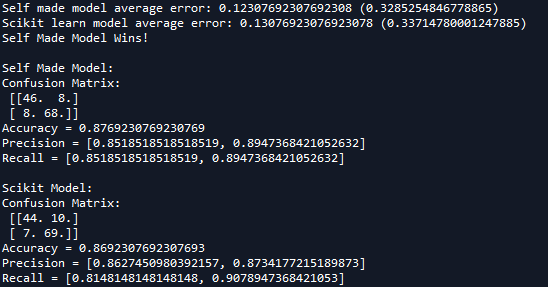
The next step is to predict the values using the model trained in the previous step. For finding the ypred vector, we dot product the xtest vector and the theta values, then sigmoid the result to get the values between 0 and 1. If the values are greater than 0.5 then they are converted to 1 else they are converted to 0.

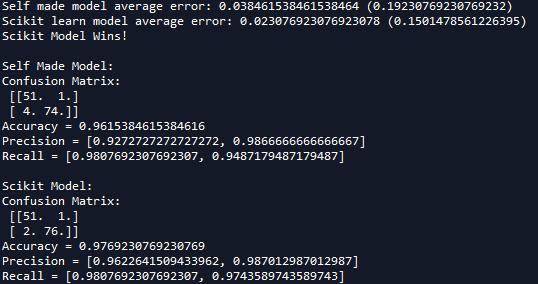
The last step is to evaluate the models. There is a function called func\_calConfusionMatrix which calculates the confusion matrix, the accuracy, the precision and the recall values. Also, we find the average error and the standard deviation of the output values.

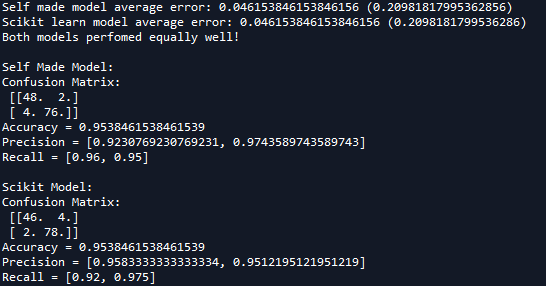
The output for different range of inputs is shown below:  
 -> 

 -> 

 -> 

Some more examples:  






**Findings:**The data should be splitted in such a way that there are enough data samples to train the model but not more or less as it may lead to underfitting or overfitting. The values of alpha and the max iterations also play an important role because they can make a model better or worse. If the value of alpha is less, than the model may take more iterations to converge or if the value of alpha is more, than there are chances that the model may jump the minimum. As we can see from the examples above, it seems that the in-built library is good for use but not perfect because in some cases the self made model goes ahead and vice versa. The confusion matrix is a very helpful entity to help evaluate the model’s performance because of its parameters such as accuracy, precision, etc.

**Conclusion:**In conclusion, we can say that logistic regression is a very good model to predict the output label values. The data should be properly scaled in order to get best results. The gradient descent is a powerful mechanism to generate minimum cost function without having to perform trial and error on each and every combination of theta values. In addition to this, the confusion matrix is a useful tool for analyzing the different aspects of the model in terms of performance, etc.

**Question 2:  
Suppose that there is a trained classifier for predicting the animal classes ( e.g., cat, dog) of a photo. The following table lists the prediction class and ground-truth class for each test image.**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Image ID** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **True class** | **C** | **C** | **C** | **C** | **C** | **D** | **D** | **D** | **D** | **D** | **D** | **D** | **D** | **M** | **M** | **M** | **M** | **M** | **M** | **M** |
| **Predicted class** | **D** | **C** | **D** | **D** | **M** | **D** | **D** | **C** | **C** | **M** | **M** | **D** | **C** | **C** | **C** | **M** | **M** | **D** | **D** | **M** |

**Notes: C, cat; D, dog; M, monkey**

**Please manually compute and report the confusion matrix and accuracy. For each of the three categories, calculate its precision and recall rates.**

**Solution:**We use confusion matrix to analyze the results of a model. For every sample, denote y as the groundtruth label and yhat as the predicted label. The confusion matrix of the above example is as follows:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Yhat = C** | **Yhat = D** | **Yhat = M** |
| **Y = C** | 1 | 3 | 1 |
| **Y = D** | 3 | 3 | 2 |
| **Y = M** | 2 | 2 | 3 |

The accuracy of the above confusion matrix can be calculated as follows:

= 0.35

The precision of the respective labels can be calculated as follows:  
 = 0.167

Similarly, the other precision rates can also be found using the same procedure,  
 = 0.375  
 = 0.5

The recall rate of the respective labels can be calculated as follows:

Similarly, the other recall rates can also be found using the same procedure,  
 = 0.375  
 = 0.428

**Question 3:  
Please write a function to calculate the confusion matrix for the prediction results of a classifier. This function should take the form:**

**def func\_calConfusionMatrix(predY, trueY)**

**where, predY is the vector of the predicted labels and trueY is the vector of true labels. This function should return accuracy, per-class precision, and per-class recall rate.**

**Please use above function in the script “main\_part1.py”, and report the confusion matrix of both logistric regression implementations.**

**Solution:**In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). We use a confusion matrix to analyze the results of a model. For every model, denote y as the groundtruth label, ycap as the predicted label.

|  |  |  |
| --- | --- | --- |
|  | **Ycap = 1** | **Ycap = 0** |
| **Y = 1** | True-Positive | False-Negative |
| **Y = 0** | False-Positive | True-Negative |

Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabeling one as another).

The function func\_calConfusionMatrix takes two matrices as its parameters. The first one is the predicted labels and the second one is the ground truth labels. It then initializes the confusion matrix based on the number of labels present in the ground truth. It traverses through both the matrices and adding the values to the respective columns in the confusion matrix. Then, we find the accuracy of the model which is the division of the sum of elements in the diagonal to the sum of all the elements in the confusion matrix. Then, we find the precision of the respective columns by dividing the label value with the sum of the elements in the corresponding column. The same procedure is repeated for the rows in order to get the recall values.

In conclusion, the confusion matrix and its parameters are a very useful tool to evaluate the performance of a machine learning model.