Classification with Logical Regression

Problem Description:

For this project I will be applying Logistic Regression to predict whether capacitors from a fabrication plant pass quality control based (QC) on two different tests. The objective is to train the system and determine its reliability with a set of 118 examples. The plot of these examples is show below where a red x is capacitor that failed QC and the green circles represent capacitors that passed QC.

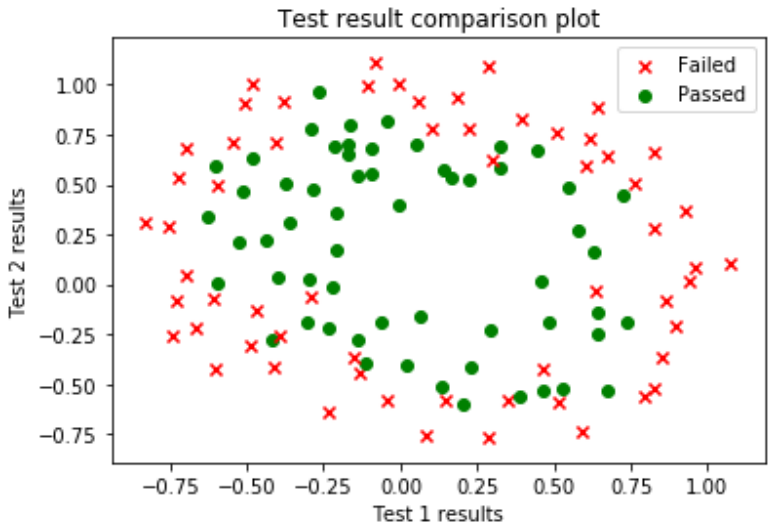

Figure 1 Test 1 vs Test 2 result plot

The data was given in randomized wat with two datasets for both training and testing with 85 examples and 33 examples in the respective datasets. The format of the dataset is as follows:

- First line: m and n, tab separated.
- Each line after that has two real numbers representing the results of the two tests, followed by a 1.0 if the capacitor passed QC and a 0.0 if it failed QC—tab separated.

Solution Approach:

The dataset was trained using the logistic regression where the code first prompts for input dataset name of training file then the features were added in the form of x1 features powered by value of 3 and x2 features powered by a value of 4. This new feature addition was saved as a final file for training. The initial values chosen for training include alpha value of 0.5 and iterations of 100,000, weights were taken as zeros initially

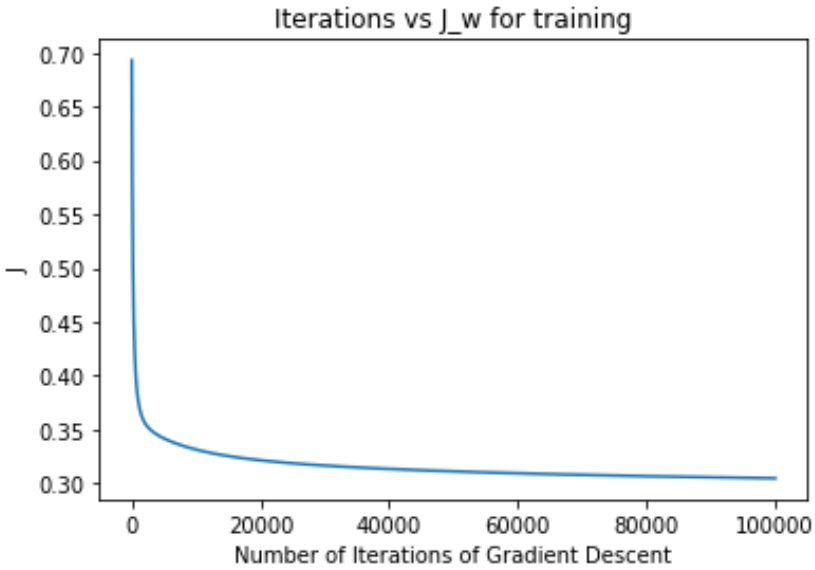

Figure 2 No. of iterations vs J value plot

and the j value was given a value of 0. After 100,000 iterations, the hypothesis and cost function along with the j values and the weights were computed and a plot was plotted between Iterations vs Jw for training as shown above. After computation, the final value of alpha remained 0.5 and the weights were computed to be,

```
Final Weights computed after training:  [[  3.90476209]
 [  1.65562789]
 [ -9.43385664]
 [  1.03847616]
 [  6.2798955 ]
 [-11.31262532]
 [ -7.97909416]
 [ 19.29342191]
 [ -0.7641728 ]
 [  2.60324433]
 [-12.44370207]
 [ 20.57528238]
 [  0.54824517]
 [ -6.30702865]
 [ -9.10553936]
 [  8.99854745]
 [-14.28637634]
 [-15.32711655]
 [-10.11198289]
 [  5.95742904]]
```

Figure 3 Final weights

For testing, the dataset was taken and additional features were added by following the same procedure with x1 features powered by value of 3 and x2 features powered by a value of 4. This new data was saved for final testing. From the weights computed during training was used and the calculation was performed to predict the classes. The J value for testing was calculated and the value was 0.47. Also, the condition of prediction less than 0.5 was set to 0 and prediction value greater than 0.5 was set to 1 for 0 to 1 plotting. Then the true positives, true negatives, false positives, and false negatives were calculated and the confusion matrix was printed. The accuracy was found out to be 0.81, the precision was 0.86, recall was 0.76 and F1 score was found to be 0.81 after testing.

```
Total number of iterations involved:  100000
Final J after training is:  0.30481072071227416
Final J after testing is 0.47262197486782836
True negative is:  14
True positive is:  13
False negative is:  4
False positive is:  2
The Confusion Matrix is:  [[14  2]
 [ 4 13]]
The accuracy in % is:  81.81818181818183
The precision in % is:  86.66666666666667
The recall in % is:  76.47058823529412
The F1 score in % is:  81.25000000000001
```

Figure 4 Output after testing

		Predicted Result: N	Predicted Result: Y
Actual Result	N	TN = 14	FP = 2
	Y	FN = 4	TP = 13

Table 1 Confusion Matrix