

Midterm Project – Logistic Regression

By: Andy Bodell

I constructed my logistic regression model by using the sigmoid function, as well as the equations to obtain the cost values, J value, and gradient descent to update the weights. To start, I read in the data and added new features to the data set. The program then asks for the name of the training set which will be 'bodell_andy_P3train.txt'. To do this, I had to create separate arrays for the x_1 and x_2 values so I could perform operations on them. I created different combinations of x_1 and x_2 by raising the values to a different power each time I iterated through the loop. Once I had the new features added, I added a column of ones to the left side of my X array so I could perform operations with the values. I also set my alpha value to .1 and the iterations to 100,000 to get my graph of J values vs. iterations to converge. Once the graph converged, I ran the model on the weights (which were initialized to zero), the x array, the y array. I ran this for 100,000 iterations to get accurate values. Once the model returned, I calculated the final H value using the sigmoid function, and then also plotted the decision boundary with the data set using the weights that were returned from the model. The program will then ask the user to input the name of the test set which will be 'bodell_andy_P3test.txt'. I then ran the model on the test set and used the sigmoid function with the weights from the training set to find the predicted Y values. These were then compared with the actual Y values from the test set and a confusion matrix was formed. Here is the data from the model running on the training set:

Final value of J on the training set is 0.34707389477369316

The final value for the learning rate is 0.1

The weights of the training set are

```
[[ 4.67831762]
 [ 2.13039967]
 [-11.24551722]
 [ 3.69941851]
 [-9.26700411]
 [ 0.33625888]
 [-10.24335132]
 [ 4.30981165]
 [-3.43958307]]
```

Final J value on the test set is [[0.34707389]]

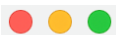
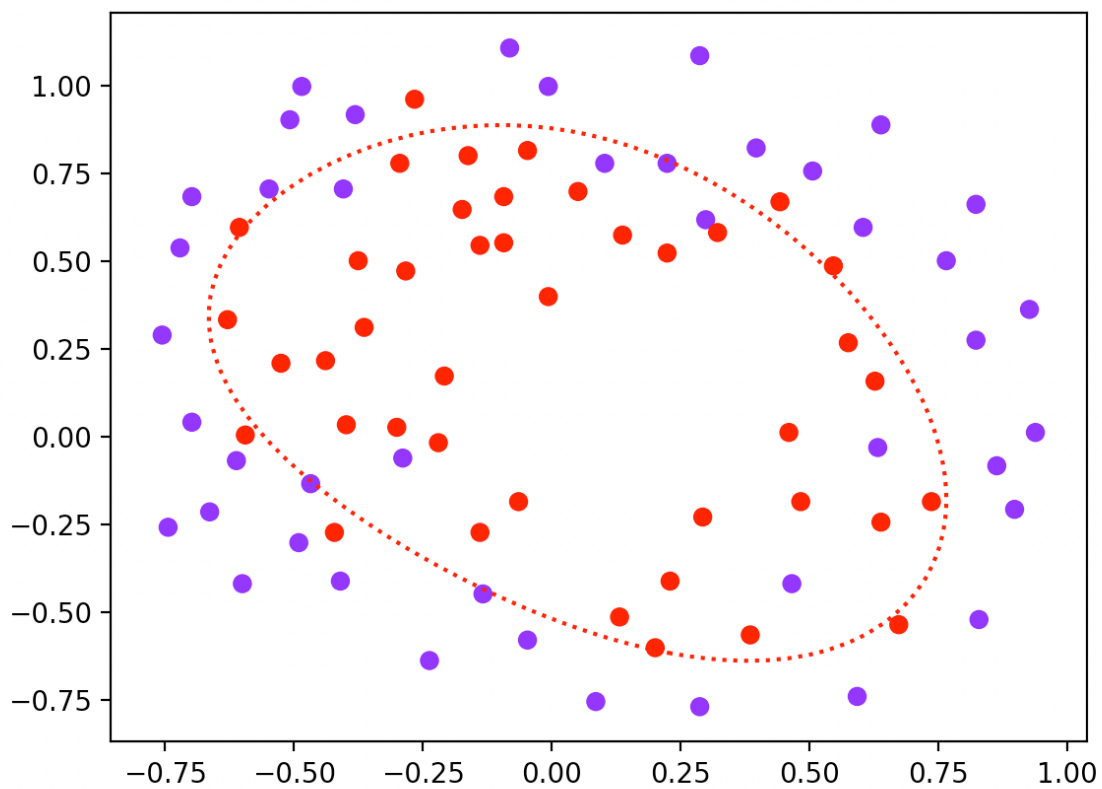
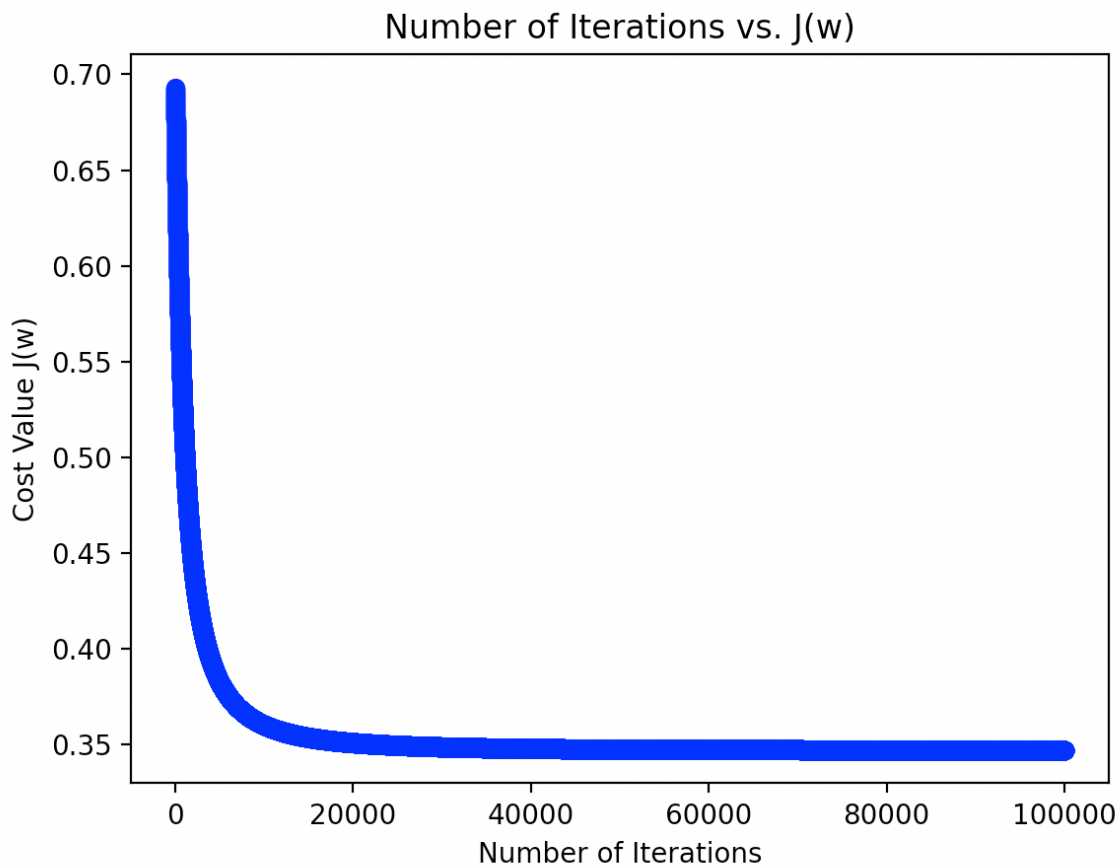


Figure 1



x=-0.073 y=0.920



Here is the data after running the model on the test set:

TP is 35

TN is 36

FP is 7

FN is 7

Accuracy is 0.8352941176470589

Precision is 0.8333333333333334

Recall is 0.8333333333333334

F1 score is 0.8333333333333334

This shows the model was fairly accurate in determining if the capacitor passed QC or not as the accuracy is around 83.5%

Along with the confusion matrix that was generated



Figure 1

