Thomas Sallurday

Professor Hodges

CPSC 6430

5 October 2020

Project 2

For my model I essentially would write a few equations, see what the F1 score was and then change the equations accordingly. The process took a while, but once I seemed to hit a ceiling of the F1 score with the number of weights that I had, I would go on and add another feature to the whole thing. I calculated the initial weights by just taking the average of each column in the data. As for calculating the new weights when using gradient descent, I had a fairly large alpha value of 0.95. I found that the larger the alpha value, the less iterations you should have to go through. I went through 2,000 iterations and got the initial J value to be 0.52. This value seemed like a pretty good value to me. Especially when considering the fact that the J value isn't an accurate predictor of how good a logistic regression algorithm is. I found that the place where I stopped for finding the weights gave me the best F1 score. My actual equation looks like a rough, uneven circle. This was the shape I was going for.

Initial Weights:

1.60419

1.76166

-2.01404

2.05935

-4.17254

3.51614

-0.259858

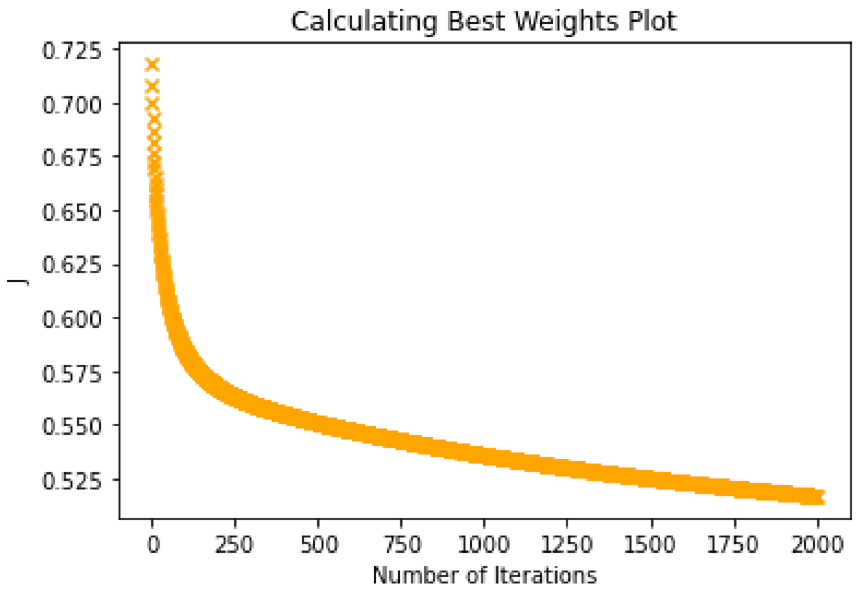
-2.76773

-8.33223

-2.12466

Final alpha value: 0.95

Final J value on test set: 0.47



Confusion Matrix:

Based on 35 inputs	Predicted NO	Predicted YES
Actual NO	TN = 17	FP = 0
Actual YES	FN = 6	TP = 12

Accuracy = 0.8285714285714286
precision = 1.0
recall = 0.6666666666666666
F1 value: 0.8

My F1 value was 0.8 exactly which is a pretty good F1 score. My precision was perfect. Accuracy was good as well at 0.82. My recall value could be better at 0.67, but the F1 value is a respectable 0.8, so my excellent precision and good accuracy make up for the lackluster recall score. To improve it, I would need to reduce the number of false negatives.