Mini Project 1 Report

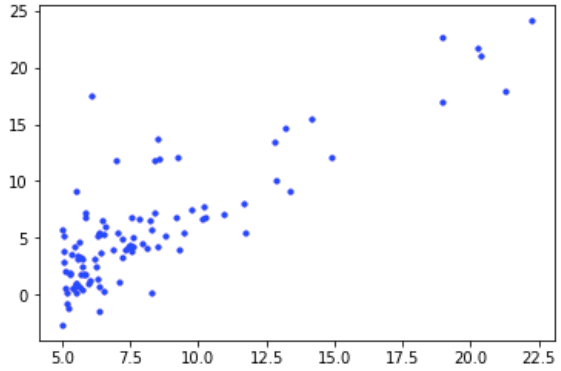
- ## Part 1: Linear Regression
  - ### Code

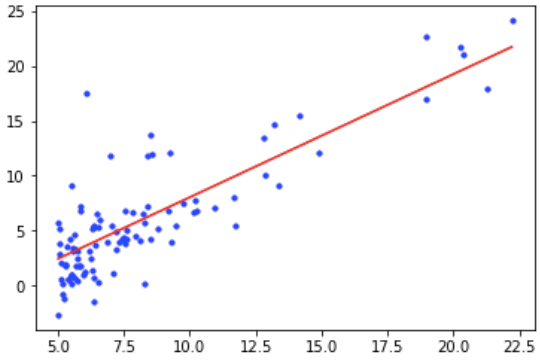
First, I'm getting data and plotting it in a graph to see what it looks like. Then I'm running 5000 iterations. In each iteration I'm calculating the cost from theta, finding the gradient and updating theta accordingly. The theta update is being done with a constant learning rate. At the end, I get the optimized theta and plot my best fit line according to the "a" and "b" obtained according to the equation y = ax + b.

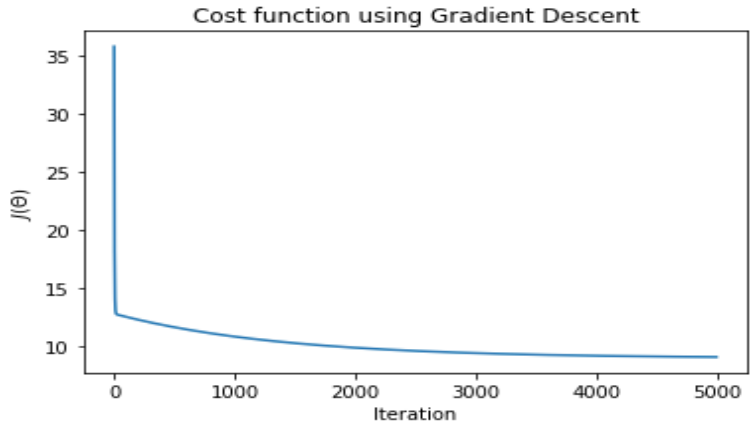
  - ### How To Run

First, go the jupyter notebook file, run the cell that imports all the required. Then go the heading that says, "Linear Regression" and start running each block one by one until there is a plot with the output line.

  - ### Plots







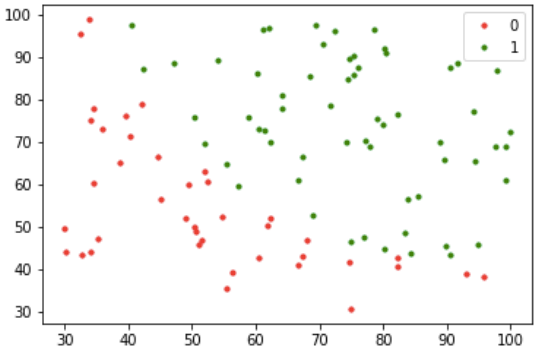
- Part 2: Logistic Regression
    - Code

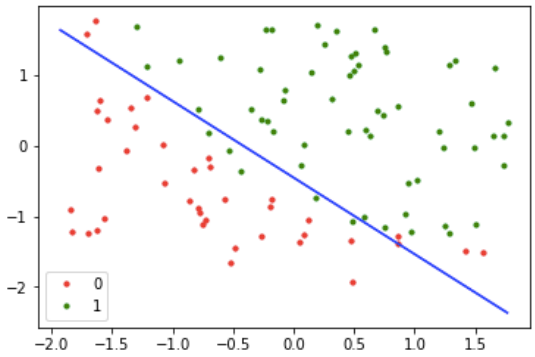
For this part, we require a sigmoid function to because we want either 1 or 0, and that is exactly what the sigmoid functions gives us according to the value we input. After defining the sigmoid function, I run 1000 iterations and in each iteration, I used the sigmoid function to give a prediction for each point in my data. Using the vectorized cost function, I calculated the error and then adjusted my theta using gradient descent at a constant learning rate. At the end, I got the optimized theta and plotted my decision boundary using it.

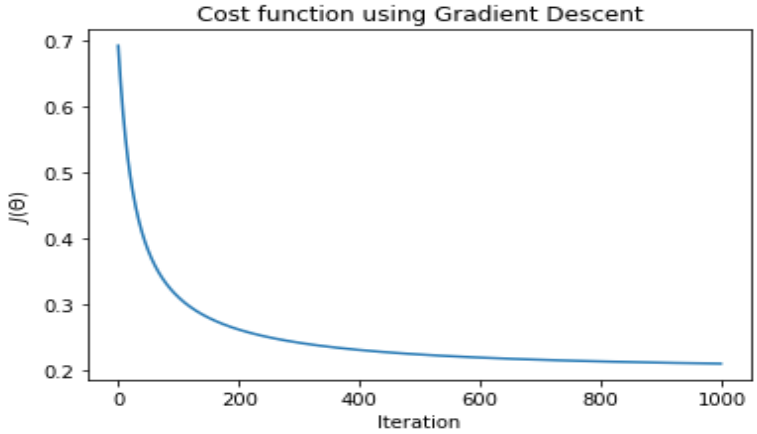
- How To Run

Go the heading that says, “Logistic Regression” and start running each block one by one from there until there is a plot with the decision boundary.

- Plots







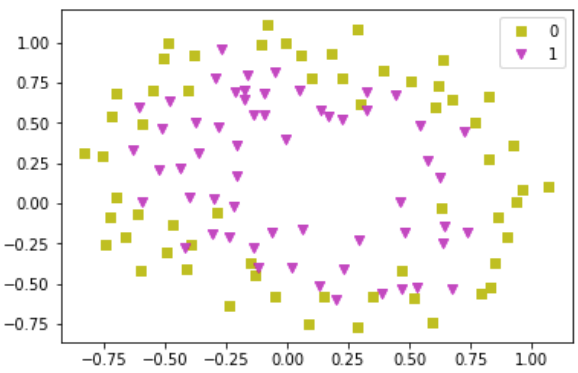
- Part 3: Regularized Logistic Regression
    - Code

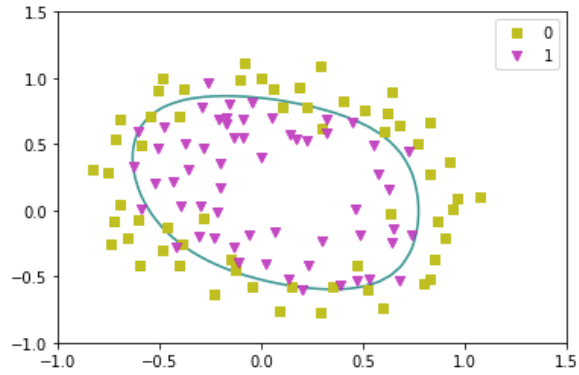
In this part, we needed to create a higher number of features to fit the data because it could no longer be separated linearly. So I created a feature mapping function for that. Then I vectorized the cost function for regularized logistic regression. In 2000 iterations, I calculated the cost and adjusted my theta using gradient descent at a constant learning rate. Then I plotted my decision boundary. This was a circular boundary and it nicely separated the two classes of data.

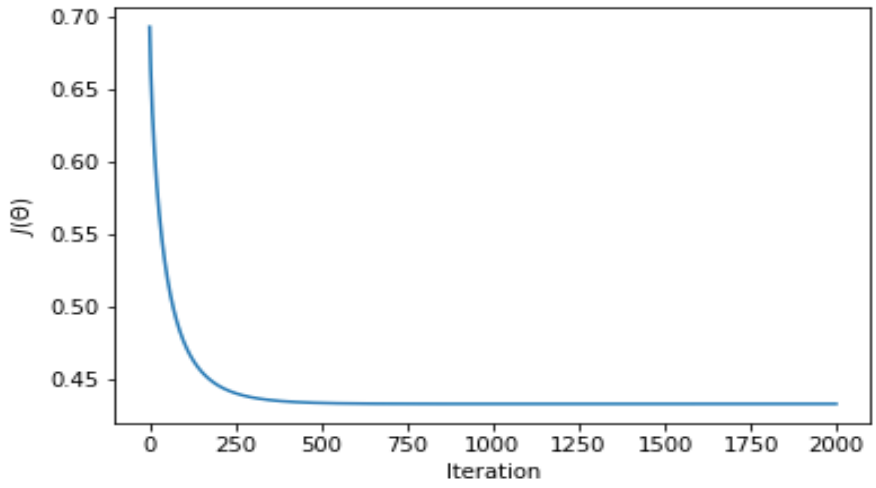
    - How To Run

Go the heading that says, “Regularized Logistic Regression” and start running each block one by one from there until there is a plot with the decision boundary.

    - Plots







- Difference b/w logistic and regularized logistic regression, and why regularization is better

The difference is regularization: it helps to avoid overfitting data. It helps us get new information that helps better classify data into different classes. This is useful to generalize data for any dataset, not just the training one.