LAB 3

ELISE MCELLHINEY

1. Linear Regression

First I normalize the inputs. I don't necessarily have to normalize the Y values, but it made my numbers come out nicer so I normalize Y as well. Usually I would want to just center Y but that makes the cost values large and difficult to work with.

1.1. Gradient Descent.

If I want to retrieve the expected values from my hypotheses then I would use:

$$E[Y] = \frac{h + mean}{std}$$

My gradient descent function outputs these as the finalized betas with the learning rate $\alpha = 0.01$ and the numbers of iterations $iters = 1000$

$$\beta = [-1.10910099e - 16, 8.78503652e - 01, -4.69166570e - 02]$$

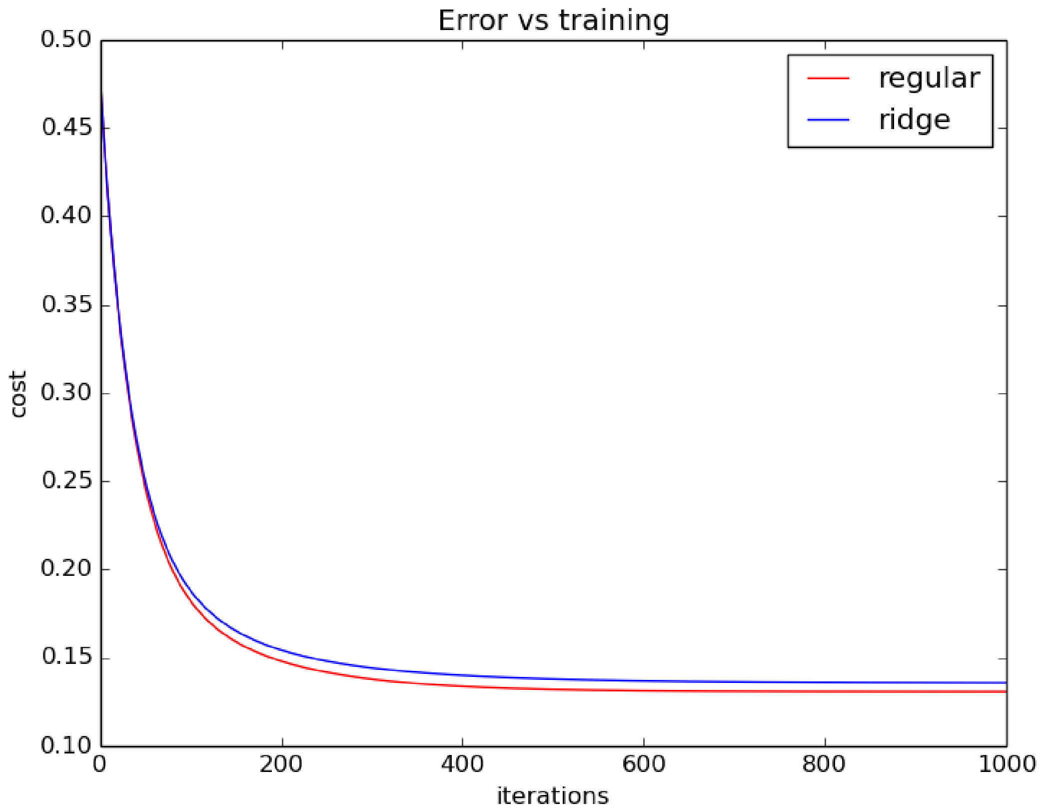
$$MLE = 0.261406739215$$

1.2. Ridge Gradient Descent.

The difference between regular gradient descent and ridge gradient descent is that ridge gradient descent penalizes large β values. My gradient descent function outputs these as the finalized betas with the learning rate $\alpha = 0.01$ and the numbers of iterations $iters = 1000$

$$\beta = [-1.10986870e - 16, 7.63060130e - 01, 1.43579162e - 02]$$

$$MLE = 0.271324713879$$



2. Logistic Regression

For Logistic regression I implemented the required functions and outputted as listed. Logistic regression provides classification instead of real-number output. My program requires that the classes are only capable of taking values of (0,1). My regression uses $\alpha = 0.01$ and $iters = 10000$

The betas produced by my gradient descent are:: $\beta = [1.71844948, 4.01290252, 3.74390304]$
The betas acquired by the optimization function were: $\beta = [1.71843665, 4.01287736, 3.74387854]$
As you can see, these are very similar. So, under the assumption that my gradient function works, my gradient descent also follows the optimization function.
My confusion matrix is as follows:

	0	1
0	34	6
1	5	55

Which shows that I have an accuracy of: 89.0% for this dataset