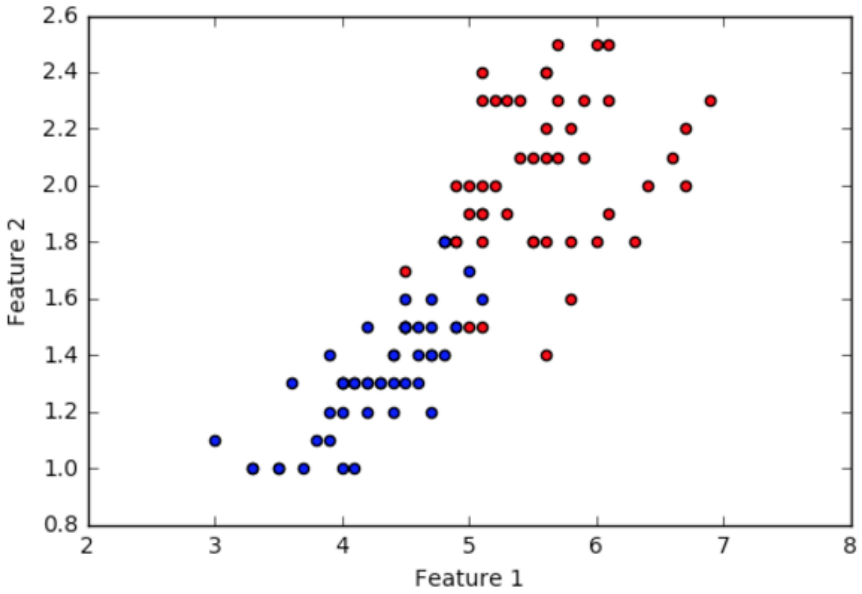
Alec Chapman
Spring 2017
Applied Machine Learning
HW3

1. **<u>Implementing Gradient Descent Logistic Regression</u>**
    a. While this data is not as clearly separated between the two classes as the example graph, this can still be seen as having two categories. Instances labelled as being in Class 0 are typically have feature values of [<5, <2].



    b. $h_\theta = 5.0645 + 0.1211x_1 - 3.5x_2$
    c. See next page

d.

Learning Rate	Cost Threshold	Accuracy
0.0001	0.01	0.5
0.0001	0.0001	0.5
0.0001	e**-7	0.94 (but extremely slow)
0.01	0.01	0.5
0.01	0.0001	0.5
0.01	e**-7	0.93

The cost threshold has an obviously powerful impact on the results of the algorithm. This makes sense, since the the cost threshold is sort of like setting a standard for the algorithm. If the cost threshold isn't low enough, the algorithm underachieves and quits before it reaches the global minimum. When it is a very small number, it's forced to keep going until it gets a cost that is pretty much as low as it will be.

The learning rate does not affect the performance, but it very clearly affects the speed. I'm using a learning rate of 0.0001 to report my hypothesis function, but realistically it is not worth the extra computing time to get that 1% increase in accuracy.

2.

a. $h_0 = 2.74 - 0.345x1 - 2.358x2$
b. Accuracy = 0.94
c. The parameters are very different from SciKit Learn's model. I'd imagine that is because it's doing a lot of "under the hood" work that I wasn't doing, such as regularizing.