**Programming Assignment-3 Logistic Regression – Part I[[1]](#footnote-1)**

**First Name1: \_\_Richu\_\_\_\_\_\_ Last Name1: \_\_\_Mathew\_\_\_**

**First Name2:\_\_\_Andrew\_\_\_\_ Last Name 2: \_\_\_\_King\_\_\_\_\_\_\_**

**Introduction**

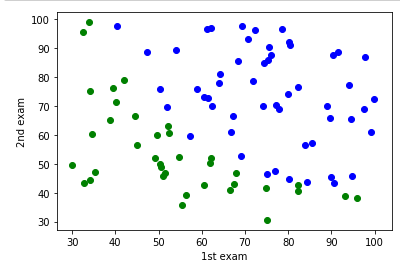
In this programming assignment, you will practice logistic regression. You will implement the logistic regression algorithm and apply it to a data set.

In Part-I, you will build a logistic regression model to predict whether a student gets admitted into a university. Suppose that you are the administrator of a university department and you want to determine each applicant's chance of admission based on their results on two exams. You have historical data from previous applicants that you can use as a training set for logistic regression. For each training example, you have the applicant's scores on two exams and the admissions decision.

Your task is to build a logistic regression (i.e., binary classification) model that estimates an applicant's probability of admission based the scores from those two exams.

* 1. **Plot the training data**

If you run the code for data loading, you can see that the data is linearly separable. Please attach the image of data visualization.



* 1. **Implement the vectorized Sigmoid function and copy and paste your code here**

**def sigmoid(Z):**

"""

Compute the sigmoid of Z

Arguments:

Z -- A scalar or numpy array of any size.

Return:

s -- sigmoid(Z)

"""

### START CODE HERE ### (≈ 1 line of code)

s = 1.0/(1+np.exp(-Z))

### END CODE HERE ###

**return** s

* 1. **Implement the loss function and gradient**

*L*(*W*) =

*α L*(*W*) =

**Solution**

loss = (1 / m) \* np.sum(-Y \* np.log(A) - (1-Y) \* np.log(1-A))

dw = (1/m) \* np.sum( (A - Y) \* X )

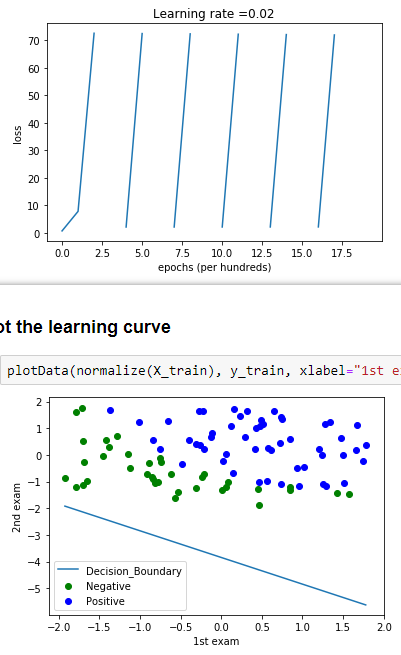
self.W -= self.alpha \* dw

* 1. **Learning the parameters based on file pa3-data1-train.csv and pa3-data1-test.csv**

**1.4.1. What will happen if you don’t perform feature scaling?**

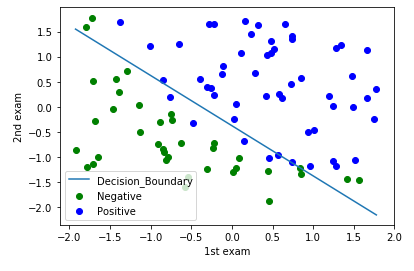
**Solution**

If you don’t perform normalization/ feature scaling, the data won’t be linearly separable also the loss being very high.



**1.4.2. Plotting the decision boundary**

After you have performed feature scaling, please run your algorithm to obtain the set of weights. Then, attach the decision boundary image in this report.



1. Part of the materials are from Dr. Andrew Ng’s course. [↑](#footnote-ref-1)