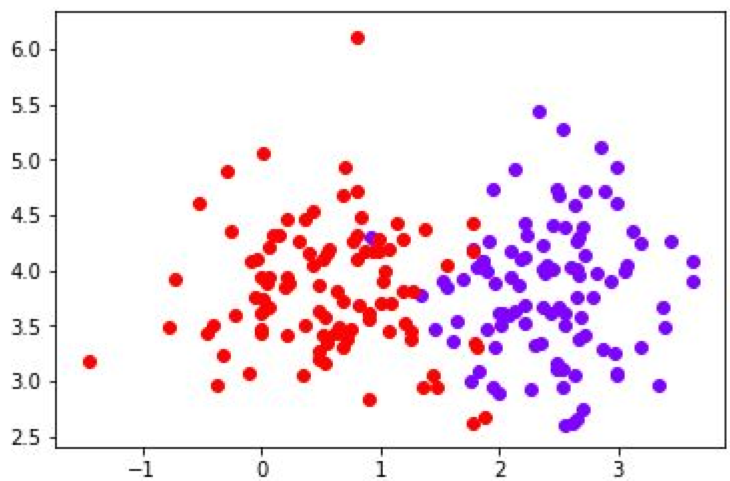
Today we will build a logistic regression binary classifier.
The model will use a simple weighted linear combination of y=w*X+b (or y=theta*X), and classifies using the sigmoid function p=1/(1+e^(-y)).

One example of an implementation can be found in:
https://gist.github.com/yusugomori/4462221
Note both sigmoid and softmax functions are used in the implementation, you can ignore softmax and focus on the sigmoid.

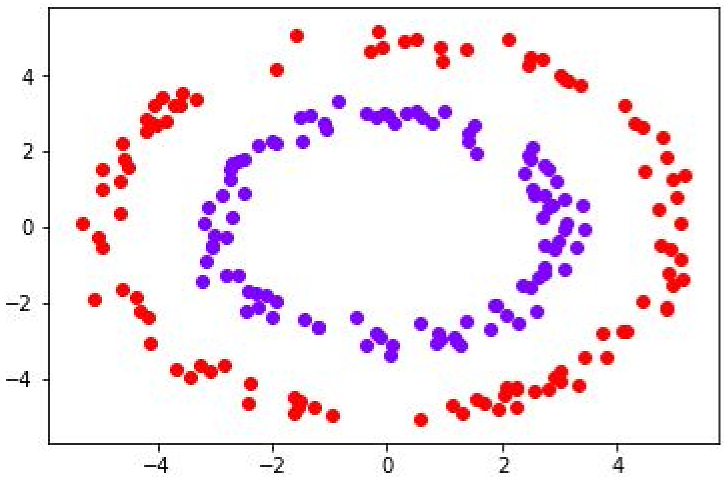
1. Create a function which generates random 2D data points with normal distribution, around a point which is not (0,0).
2. Generate your data by choosing two points and generating two groups of random normal distribution data points around them (try to make the variance of the random distribution large enough so some of the data will overlap) e.g:



3. Store the data point in X and the labels in y (0 or 1 depends on which group this points belong to).
4. Create a function which calculates the sigmoid function
5. Create a function which returns the prediction of logic (result>0.5)
6. Create a function which calculates the negative log likelihood function (which is the cost/loss function)
7. Create a function which calculates the gradient of the loss function (gradient of the combination of the functions in 4,6). Try to calculate it by hand first!

8. Create the training function for the logistic regression model, which starts from some random values for the parameters of theta, and performs gradient descent to minimize the loss.
9. Train your classifier with enough iterations and see if it manages to classify the data correctly (try different learning rates, epochs and initializations for theta), split your data into train and test sets and print out the precision of prediction on each after training.

10**. Create another set of data consisting of points from two circles with different radii and some noise, and try to create a classifier for the data



Good Luck!