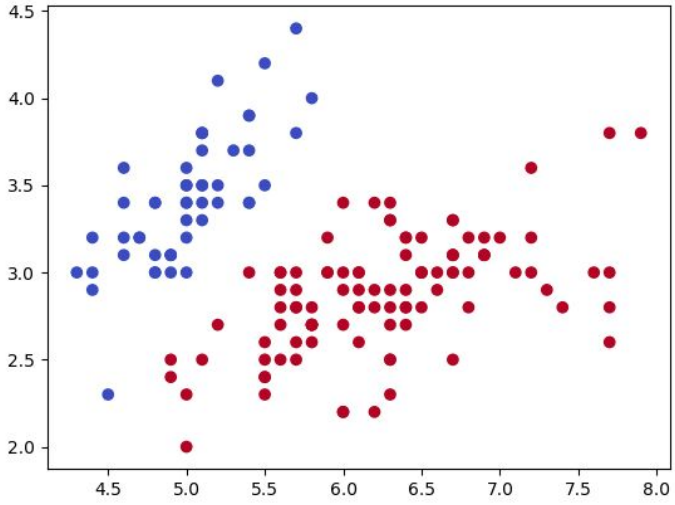
Today we will build a logistic regression binary classifier.
One example of an implementation can be found in:
https://gist.github.com/yusugomori/4462221

First, dealing with data:

1. Read iris dataset into a pandas dataframe
2. Use the first two columns as X
3. Labels will be decided this way:
    a. If the last column value is 'Iris-setosa" - 1
    b. Else - 0
4. Create visualization of the data points



Secondly, creating the model:.

1. Create a function which calculates the logit function

$$g(z) = \frac{1}{1 + e^{-z}}$$

2. Create a function which calculates the negative log likelihood function (which is the cost/loss function)

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1 - y)^T \log(1 - h)\right)$$

3. Create a function which calculates the gradient :

$$\frac{\partial J(\theta)}{\partial \theta_j} = \frac{1}{m} \sum_{i=1}^{m} (h(x^i) - y^i)\ x_j^i$$

4. Create a function for train-test split by a percentage
5. Create the training function (fit 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.
6. 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.
7. Print the best train and test accuracy.

Good Luck!