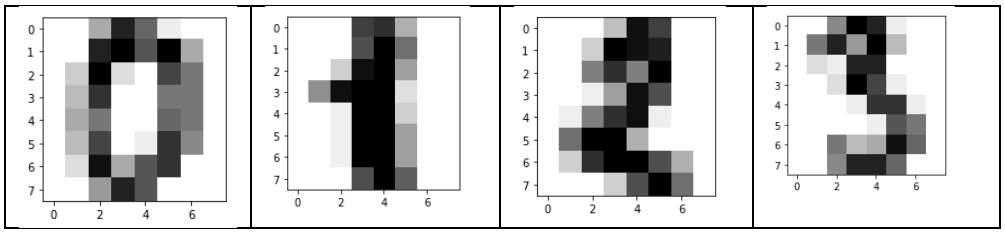
Logistic Regression for Image Classification

Question 1 – Logistic Regression Models for Binary Image Classification

In the practical lab folder we will be developing a Logistic Regression model that will perform classification on a basic image dataset, called the Digits dataset. Each sample in the dataset is an 8x8 image representing a handwritten digit (see below for some samples). There are 1797 8x8 images. Each pixel is represented by an integer in the range 0 to 16, indicating varying levels of black. As logistic regression is a binary classifier we will be building a model to differentiate between two different digits. (This is a very basic image classification problem. We will be covering much more complex and realistic image classification tasks later in the module).



Instructions:

On Blackboard you will find a template code file. When looking through the code you should notice just how similar it is in structure to the MLR code you completed in last week's lab. Complete the code for the logistic regression classifier as follows (please note this is not intended to be a vectorised implementation):

1) Complete the function called logistic. The objective of this function is to take in a NumPy array and to pass it through the sigmoid function and return the results. The NumPy function np.exp should be useful in creating this function.

2) Complete the function *hypothesisLogistic* so that it returns the correct predicted value. Remember this function takes in the training data, the current coefficients and the bias and should return the value predicted by our logistic classifier. This will be very similar to what you wrote for the MLR exercise.

3) In the gradient_descent_log function complete the line of code that calculates the cross entropy error. You should see this value decrease every time you iterate around the gradient descent loop.

4) Complete the calculateAccuracy function. Remember logistic regression is a probabilistic classifier that produces a prediction in the range 0-1 . How should you convert it into a final classification value?