Problem 1:

The file `X_data` contains 200 numbers drawn randomly from two probability density functions. Plot an estimate of each probability density function on the same figure. Estimate the expectation of each. To which of the two functions does the number 9.9 more likely belong? Explain your thought process. Include your well-commented codes. Do not use inbuilt toolbox functions.

Problem 2:

The variables `X_1` and `X_2` each consist of 10 data points of dimension $\mathbb{R}^2$ drawn from two classes (1 and 2). Plot the data points from Class 1 and Class 2 on the same figure but distinguish between the points from each class using a legend. Now find a separating hyperplane (line) using:

(a) the perceptron algorithm; and

(b) a support vector machine algorithm.

For each algorithm, superimpose your separating hyperplane on the original plot of the data points. Discuss the difference between the hyperplanes resulting from each algorithm. Include your well-commented codes. Do not use inbuilt toolbox functions *except* optimization solvers such as `quadprog` and `fmincon`.

Problem 3:

Build a neural network (a multi-layer perceptron) to recognize handwritten images on the MNIST database. While the database consists of images of all handwritten digits from 1 to 9, for this assignment, we only consider images of hand written digits from 1 to 3. The attached folder `Reduced_MNIST_data` comprises `train` and `test` folders. The `train` folder contain three subfolders each holding 1000 images corresponding to the digits 1, 2 and 3. The `test` folder contain three subfolders holding 200 images for each of the three digits. Each image is of dimension (28×28). Your neural network is expected to take an image of a hand-written numerical digit on the `test` data and classify it into one of 3 digits.

To train the network, use analytic gradient expressions to implement a gradient descent or stochastic gradient descent scheme. You may use any suitable loss and activation functions. Your report should (i) indicate what loss and activation functions you used; (ii) explain your training process; (iii) plot your loss function against gradient descent iteration number; (iv) draw a confusion chart to show your classification accuracy for each of the three classes; and (v) indicate your average classification accuracy. In addition, submit your neural network as a `.m` function that will take in an image input vector and output the digit corresponding to the image. Your neural network is expected to average classification accuracy of at least 80% on the test data. You will receive a bonus point for every 10% your average accuracy exceeds 80%. Do not use inbuilt toolbox functions *except* optimization solvers such as `fmincon`.