# Pattern Recognition (Spring 2021): Assignment 2 Prof. Zanibbi

**Due Date: 11:59pm, Sunday Feb. 28**

The assignment is out of 50 points. **Submit your write up (.pdf file) and python programs in a .zip file (`code.zip`) as separate files through the assignment drop box on MyCourses.**

**Questions:**

1. (20) Implement a *maximum a posteriori (MAP)* classifier, using Gaussian distributions to estimate class-conditional densities. Create a program `a2map.py` that reads in the `data.npy` file from Assignment 1, and creates a plot showing 1) the training data, 2) the decision boundary separating class 0 and 1, and 3) the classification regions.

   The program should save the plot as an image file. Also, have your program write the classification rate (% correct) and confusion matrix for the training set on the terminal (standard output) - use the same organization for the confusion matrix as shown in Assignment 1.

   **You must create your own functions to compute mean vectors, covariance matrices, class priors and the class-conditional (Gaussian) feature densities - do not use a library to compute these values. The decision boundary should also not be created using a library call.**

   **In the write-up,** provide the plots and confusion matrices produced by your program, along with your comments on the results. **Describe and explain** the similarities and differences between your MAP classifier results, and the results obtained using the least-squares (linear) and k-nearest-neighbor classifiers. You may reuse the plots you created from assignment 2 to make it easier to compare the classification models.

2. (20) Create a second program, `a2cost.py`. This program will use your (*approximated*) Bayesian classifier from Question 1, but with different cost functions.

   Using the provided `nuts_bolts.csv` file from Duin et al.'s "Classification, Parameter Estimation and State Estimation," have the program create the plots shown in Figure 2.5a) and c) (i.e. using the cost function on page 20, and using the the *0-1 loss* cost function). Then, modify the prior probability of the scrap class as described in Duin on page 21 (Example 2.4) to produce Figure 2.5 b).

   **The computations for classification and decision boundary locations should be produced using your own code, and not a library.** Have your program save the generated figure images to disk as files.

   **In the write-up**, provide the three plots created by your program along with the recognition rate and confusion matrix produced by each classifier. Then comment on the differences between the different classification models.

**Bonus (5 points):** Add an implementation of the classifier with a rejection option as defined in Section 2.2. (Example 2.6), with the cost function shown in Table 2.3. Reproduce the plot in Figure 2.11, and then include the plot in your write-up along with the confusion matrix obtained for the training sample, and then comment on the location of the reject region, and what causes it.

3. (10) Assume that the number of scrap objects in Figure 2.2 in "Classification, Parameter Estimation..." is doubled, while keeping the same number of other objects, but **without** affecting the class-conditional density shapes (i.e., the class-conditional means and covariance matrices remain the same). After making the necessary adjustments to prior probabilities, revise the cost matrix in Table 2.2 so that the **decision boundaries do not change** with these additional scrap objects added to the training set.