NAME: _____

Instructions: Each problem should usually include:

- the final answer placed in the provided box, and
- the detailed derivation of your answer (text and equations).

Your solutions must be submitted via Canvas by 11:59 pm on the due date as a single .pdf file with all pages oriented in the same direction. You should use an annotating program, scan work done with pencil and paper, or typeset your solutions using Latex. Whatever method you use it should be neat, legible, and well-organized. You can insert pages if needed but maintain the question ordering. Camera-phone images will not be graded.

This problem set consists of design problems only. Your solutions must be submitted via Canvas by 11:59 pm on the due date as a single zip file containing the julia files for each problem (`ps4p1.jl` and `ps4p2.jl`).

Note: This assignment focuses on the material in lectures 9-11.

---

1. **Design Problem** Implement kernel estimation (Parzen windowing) and use it in a binary classifier. Place all your code in the file `ps4p1.jl`.

   (a) Implement a julia function

   `function kernel(x, data, h)`

   To estimate p(x) at a point x (D-vector) given a data matrix (N x D) and a width h using a Gaussian kernel (window function).

   (b) Using your kernel to estimate the class conditional densities, and a maximum-likelihood estimate of the class prior probabilities, implement a function to perform classification of new samples x, given the data and h, using a likelihood-ratio test (compare the ratio of the class conditionals to the ratio of the priors).

   `function classify(x, data, h)`

   (c) Read the data file `ps4p1.h5` containing an 1000×2 matrix (group name "data") representing a training set for a binary classification problem with a single real feature x. Perform leave-one-out cross validation to select the optimal value of h. Enter the optimal h found.

   (d) Plot the cross-validation error as a function of h from part c.

2. **Design Problem** k-Nearest-Neighbor classifier. Place all your code in the file `ps4p2.jl`.

   (a) Implement k-Nearest-Neighbor classifier as a julia function

   ```
   function knn(x, data, k)
   ```

   where x are the new values to classify, data is the training data, and k is the number of neighbors to use.

   (b) Read the data file `ps4p1.h5` from problem 1. Perform leave-one-out cross validation to select the optimal value of k. Enter the optimal k found.

   (c) Plot the cross-validation error as a function of k from part c.