**Building a One Dimensional Classifier**
**Dr. Thomas B. Kinsman**

Homework is to be programmed only in one of the following languages. No other languages will be accepted. Please limit yourself exclusively to: Python, Matlab, Java, or R.

Assume that the grader has no knowledge of the language or API calls, but can read comments. Use prolific comments before each section of code, (or complicated function calls) to explain what the code does, and why you are using it. Put your name and date in the comments at the heading of the program.

Hand in your results, and the commented code, in the associated dropbox.

Submit two files:
a. HWNN_<LASTNAME>_<Firstname>_write_up.pdf,
with your results, graphs, and the answers to questions in it.
b. HWNN_<LASTNAME>_<Firstname>_program.extension, with your program in it.

You can look over each other's shoulders, and at each other's work, but do you own work for the submissions. Let me know whom you worked with. Do not hand in copies of each other's code.

CAUTION: The bin size here is to the nearest half mile per hour. This is sometimes different from the previous homework.

Presume that you break the speeders into two groups: one group that is ≤ the threshold, and a second group that is > the threshold. Speeds that are ≤ the threshold go into the first group of cars.

1. You will be provided with a file of driver speeds, several other attributes, and if they are trying to drive aggressively. (See the column headers.) The aggressiveness was based on an officer painstakingly interviewing the drivers.

    a. ( ½ ) Considering that we are trying to maximize public *safety on the roads*, how would you break a tie if two different speed thresholds have the same lowest misclassification rate? How would you set your threshold to the lower or the faster speed? Why?

    b. ( ½ ) Imagine that you are trying to maximize how much trust the public has in the police officers, how would you break a tie if two different speed thresholds have the same lowest misclassification rate? Why?

    c. ( 3 ) Define a cost function such that false alarms are just as bad as a miss.
    So, the cost function will be equal to (number false alarms + number of missed speeders).

    Using the techniques covered in class, write a program to find a threshold for a police officer to set their laser speed detector at so that it beeps in such a way that it minimizes this cost function.
    In case of ties, maximize the public's trust that a police officer is not pulling people over for the fun of it. (Use the higher threshold.)

    Here, I want you to round the speeds to the nearest 0.5 mph, sort them, and then try the speeds from slowest to the fastest.

    What threshold value did you compute as the best threshold? (To the nearest 0.5 mph)

d. ( 1 ) Suppose that we want a cost function such that false alarms are two times worse then a miss.
So, the cost function will be equal to (2 x the number false alarms + 1 x number of missed speeders),
*or maybe* it is (1 x the number false alarms + 2 x number of missed speeders).
Guess how the threshold will move. What do you guess? Up or down? Faster or slower?

*Temporarily* change your program to minimize the cost function, and check your guess. How did the threshold change? ***Temporarily*** modify the program to minimize this cost function.

What *new* threshold value did you compute for this new cost function?
(To the nearest 0.5 mph) Does this change make sense? Why or why not?

e. ( ½ ) Decompose this *temporary* cost function in terms of an objective function and regularization.
What is the regularization being used here? What does the regularization penalize?
Are there any issues with the relative amount of regularization here?

f. ( ¼ ) Change your program back to using the first cost function.
For the given training data, how many aggressive drivers does this let through for the given data set?

g. ( ¼ ) For the given training data, how many non-reckless drivers would be pulled over?

h. ( ½ ) How does this value compare to the value you found using Otsu's method in the previous homework?

i. (1) Plot the cost function as a function of the threshold used.
Label all axes.

j. ( 1 ) Generate a receiver-operator (ROC) curve for this training data.
Plot it, and put the location of the any thresholds on the ROC curve.
Label the X and Y axes correctly.

Try to make the axes square if possible, so that relative slopes can be compared.

Circle any point (or points) on the ROC curve with the lowest cost function.
Caution, there may be more then one of them.

k. (1) **Conclusion:** Write up what you learned here using at least two paragraphs.
How might you use a one-dimensional classifier with multi-dimensional data?
Was there anything particularly challenging?
Did anything go wrong? Provide evidence of learning.