ML Foundations
HW 2: Univariate and Multi-Variate Discriminants

Item(s) to turn in: jupyter notebook, classifiers.py, discriminants.py and pdfs of each file (notebook, and python files)

Approved Libraries: numpy, pandas, matplotlib

---

1. Your first goal is to fill in the Gaussian discriminant class (univariate) and test it. You can look at the lecture slides for the calculation of $g_i(x)$. (5 pts)

   For testing:

   - Create two numpy arrays (resting, stressed), where resting data has a mean of 60 and stdev. of 5 and stressed has a mean of 100 and stdev. of 5.
   - Fit a uni-variate Guassian to each array
   - Plot the resulting discriminant values for inputs (x) 40 to 120 and highlight the decision boundary (where they cross)

   Short Answer:

   1. State why the two disciminants cross at 80 and why there is a parabolic shape to each discriminant plot

2. Your first goal is to fill in the Gaussian discriminant class (Multivariate) and test it. You can look at the lecture slides for the calculation of $g_i(x)$. (5 pts)

   For testing:

   - Create two numpy arrays (resting, stressed) that have two features each. $\mu_{rest} = [60, 10], \Sigma_{rest} = [[20, 100], [100, 20]]$. $\mu_{stress} = [100, 80], \Sigma_{stress} = [[50, 20], [20, 50]]$
   - Fit a multivariate gaussian discriminant to each array
   - Plot the resulting discriminant values for inputs (x) [20,20] to [120,120] and highlight the decision boundary (where they cross)

3. Your next goal is to fill in the Discriminant Classifier class. The set_classes function may be useful for you to use/implement without having to refit data each time. (10 pts)

   Key items to think about:

- Is your code usable for more than 2 classes?
- How do you pool variances together? Hint: look at lecture slides

For testing:

- Using the multivariate discriminants you created in problem 2, plot the decisions the discriminant classifier outputs for inputs (x) [20,20] to [120,120]. You should see similar boundaries to problem 2.
- Now we need to connect theory to our implementation. First, create a pooled covariance matrix (shared class covariance), replot your decisions and briefly discuss how the boundary changed.
- Next, (still using the pooled covariance matrix) change the priors of a class and see what happens to the decision plots.

4. Answer the following questions in MarkDown cells in your Jupyter Notebook: (10 pts)

   1. Explain why we assume I.I.D. for Maximum Likelihood Estimation (MLE)
   2. Prove the Maximum Likelihood Estimation for a Bernulli Distribution is $p_0 = \sum_{i=1}^{N} x_i/N$. *Hint* The lecture slides have the distribution equation for you.
   3. We talked about Maximium Likelihood Estimation for various distributions from an abstract sense; thus, lets create a concrete problem. I have a bag containing *Red* and *Green* balls. Let Θ be the number of green balls and the total number of balls in the bag is 3. Assume $X \sim Bernuoilli(\Theta/3)$ and you draw the following samples $[x_1 = 1, x_2 = 0, x_3 = 0, x_4 = 1]$ (We observed 2 green balls). For each value of Θ, fill in the resulting joint probability in the following table: *Hint* The distribution is $P(x) = \begin{cases} \Theta/3 \\ 1 - \Theta/3 \end{cases}$

Θ	Value
0	
1	
2	
3	