# DATA 221
# Homework 2
**Trimble**
Due: 11:59pm Friday 2023-04-07

1. **Naive Bayesian Spam Classifier**

   This problem asks you to build a function to estimate the (posterior) probability that an SMS message is spam based on the words (or some subset of the words) that it contains.

   The UCI "SMS Spam Collection Dataset" submitted by Almeida and Hidalgo, is a collection of 5000 text messages, 13% of which are labeled as spam. Tokenize and count the word usage for the spam messages and the word usages for the ham messages.

   `https://archive.ics.uci.edu/ml/datasets/sms+spam+collection`

   Construct a function that scores new text messages by estimating $\frac{P(spam)}{P(ham)}$ by taking some function of the empirical word frequencies in the dataset $P(word|ham)$ and $P(word|spam)$

   This is an ill-posed problem. We have to decide how to handle words that are absent from one (or the other) dataset, words with small numbers of occurrances, and decide how many words to use. Describe your choices briefly.

   To prevent our model from gettig carried away, let's limit the influence of any word to a factor of $\pm$ log(100) in log-odds score: a message of 4 words that only appear in the spam corpus will get a score of 100,000,000:1, and an utterance of 2 words that appear only in the ham corpus would get 1:10,000.

   (a) Split the UCI corpus into a training and a testing dataset, say, 90/10, train the model on the 90%.

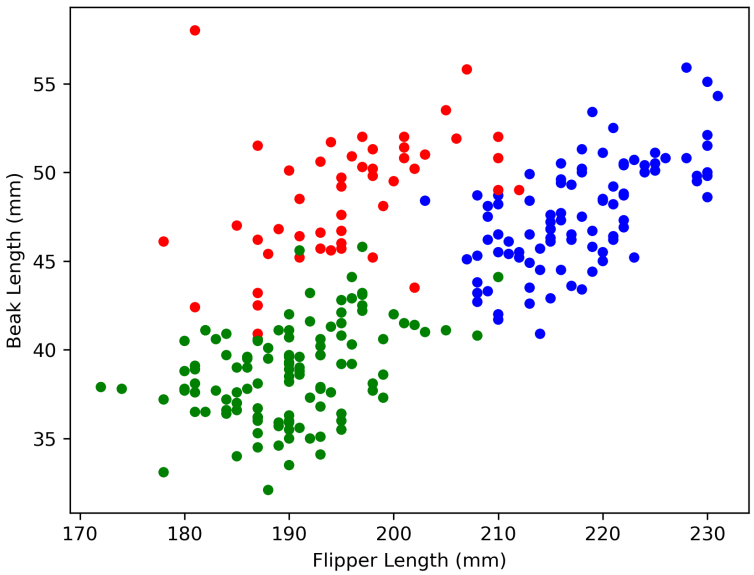
   (b) Evaluate and report the confusion matrix after running the classifer on the testing subset (about 50 spam and 400 non-spam).

   (c) Score all of the messages in the "testing" corpus compiled by Mohammed Noor Hassan in

   `https://github.com/mohammadnoorulhasan/sms-spam-prediction/blob/master/Problem/Testing.csv`

   and plot the histogram of log-odds-scores of messages (disaggregated by Spam /not spam). Plot the histogram of probabilities for these messages.

   Example code for tokenization will be provided. You might recognize this as a variant of the loaded dice problem; words are the outcome of a dice with 50,000 sides.




2. **Linear Regression, Logistic Regression, and Linear Discriminant Analysis** This dataset has two ”label” variables (sex and species) and four numerical ”features.” We will try to classify penguins by species using three techniques that look at linear combinations of the feature vectors.

   (a) As before, split the dataset into training and testing subsets. Maybe 80/20 this time, we don’t want to run out of penguins.

   (b) Find logistic regression coefficents to classify penguins by sex using the four four-dimensional X (flipper length, beak width, beak length, and mass).

   (c) Classify the test set by sex and report the confusion matrix.

   (d) Find logistic regression coefficents for the indicator variables for species identity against the four-dimensional X. Plot the decision boundaries between the classes implied by the regression coefficients on top of the scatter plot. This requires a little bit of algebra.

   (e) Find logistic regression coefficents for the indicator variables for species identity against the four-dimensional X + sex indicator variable.

   (f) Classify the test set by species and report the confusion matrix for both of the classification methods above.

   Here you can either plot the boundaries by finding the equations for the boundary or, if you find it easier, evaluate a classifier at a few hundred points on a 2d grid and plot a symbol on the graph indicating which regions of X get which classification; you can solve this with math or you can solve it numerically.