Franco, Jonsoon, Tod, Eric

Exercise 1: PR() instead of just P for the probability in Mathematical formation

Exercise 2: see Lab6\_ex2.pdf for math… The likelihood of am email considered spam goes up as a number of words that appear in the spam section goes up. When words that appear in the ham section, we can see the probability will go down that it is spam. Just by removing the "work" word. The probability went from 90% to 94.7%. It also makes sense when we remove the "cheap" word that the property went down again because there is less words to make a good guess.

Exercise 3: Spam is a terrible thing. It would be nice if we had a machine that can go through our inboxes and tell us what spam is and what is not spam. We could do that… but how does one consider something spam and what is considered ham (not a spam email)? We can do this with categories. For spam, it can either be spam or ham. Now, we need to training data that can we can put into the two categories. Grab a bunch of emails and manually sort them. Put the spam in the spam section and the real emails in the ham section.

Once you got that all figured out. You are going to need some kind of formula to compute for future emails. Bayes (formula not given) is a great way to do this. This uses prior probabilities to produce a posterior probability. So basically, what’s going on is that we are determining something is spam or not based of the what the training emails where. We can do better; we don’t even need to do that. Since all hypotheses are priori equally likely. Then we just really need to do this. Cml === argmax P(X|c). We also want to consider that we want our emails to be independent from each other, so we need to use Naïve Bayes theorem.

One more thing we should also consider is not everything is going to make it in the training data. So we are going to need something to “smooth out” that missing data. We can use +1 or some other constant (alpha). Another trick to know is to instead of multiplying the probabilities is to just log them and add them all together.

Exercise 4 : see Taxis.py