Team members:

- James Ho
- Nicholas Schindler

Data Set: https://www.kaggle.com/datasets/uciml/sms-spam-collection-dataset

Topic:
We are interested in using SVM to filter real texts from spam based on the content of the messages. Spam is a part of everyday life – we all get it, and at times the overwhelming number of messages we have to sort through can feel overwhelming. What's interesting about this specific data set is how all of the variables are consolidated into a single field. This will force us to do extra work to extract the data. However, while extracting the data, we can also format it in such a way that it is optimal for analysis.

The data that we will be using is a set of SMS tagged messages that have been collected for spam research. It contains 5,574 messages that have been tagged either legitimate(ham) or spam. What makes this data set interesting is that the data consists of raw text labeled either spam or ham, and from there we can use the data to machine to learn what words would be most common in a spam text. This will allow us to figure out if a text is spam or not based on the words.

The data set was generated from the following data sources:

- https://www.grumbletext.co.uk/
- https://etheses.bham.ac.uk/id/eprint/253/1/Tagg09PhD.pdf
- https://archive.ics.uci.edu/dataset/228/sms+spam+collection

This data is in a very difficult format. It is built in such a way that there are only two columns: the classifier "ham" or "spam", and the string that is the message. We will have to create a parser to go through the data and generate our own statistics based on which words are often used. However, it's not that simple. Many of the texts have misspelled words and/or special characters, as well as punctuation. We will have to do significant work to get usable data from the parser. On the surface, our data analysis has nothing to do with the content of the course. The benefit is that this forces us to consider the nature of SVM and the type of data that it excels at analyzing. We will have to seriously consider how the SVM algorithm works to set our data up for analysis.

Timeline and Responsibilities:

- Create algorithm to parse texts and find often-used words
    - Target date: 21 Nov

    - Programmed by: Nick Schindler
    - Reviewed by: James Ho
- Use to build a dataset of words that are used for spam vs. real texts
    - Target date: 28 Nov
    - Programmed by: James Ho
    - Reviewed by: Nick Schinder
- Create an SVM using the dataset that we generated
    - Target date: 5 Dec
    - Programmed by: Nick Schindler
    - Reviewed by: James Ho
- Analyze the SVM data
    - Target date: 10 Dec
    - Programmed by: James Ho
    - Reviewed by: Nick Schinder
- Submit report
    - Target date: 12 Dec
    - Submitted by: Both