PROBLEM FRAMING COURSE
Sarah Griffioen

Exercise 1: Start Clearly and Simply
I want the ML model to identify whether an email in Gmail is "important."

Exercise 2: Your Ideal Outcome
My ideal outcome is to save the Gmail user time and keep them on top of their tasks. Anyone with an email account knows that so many emails can come in each day, and it is difficult to sift through all of them. However, they need to be sorted through because some of those emails are very important and require action in the near future. If I could identify which messages are the most important, I could save the user a lot of time and make sure that they stay on top of all of their tasks.

Exercise 3: Your Success Metrics
A success metric is user clicks and responses to emails. Success means that users click on and respond to 80% of the "important" email messages within 24 hours of receiving them. Failure means that users click on and respond to less than 80% of the "important" email messages within 24 hours of receiving them.

Exercise 4: Your Output
The output is to predict whether the email message will be important to the user. The ideal outcome is to sort the email messages by putting the most important ones on the top with an "important" tag on them.

Exercise 5: Using the Output
I will predict if an email is "important" when the email message is received. The outcome will help determine where the email message goes in the sorting order based on its level of importance.

Exercise 6: Your Heuristics
Consider contacts who have sent the most email to the user and have received responses from the user in the past. It should not only look at the contacts who send the most email to the user because these could be spam senders that the user never replies to since it is not an important message. From this information, assume that new email messages that are sent by the contacts who have sent the most email and received the most responses will be the "important" messages.

Exercise 7a: Your Problem, Formulated as an ML Problem
My problem is best framed as a three-class, single-label classification, which predicts whether an incoming email message will be in one of three classes – {very important, somewhat important, not important}.

Exercise 7b: Cast your Problem as a Simpler Problem
I will predict whether an incoming email message is "important" or not (binary classification).

Exercise 8: Design your Data for the Model

Subject	Sender	Time Received	Message response count from the past 2 weeks	Output
Plans for Graduation	Bonnie Griffioen	2019-03-2 08:00	35	Very important
Play for chapel TOMORROW!	Paul Ryan	2019-02-13 09:36	13	Very important
Grades for CS344	Keith VanderLinden	2019-01-21 02:47	10	Very important
Department of Music News	Department of Music	2019-02-6 12:53	0	Somewhat important
List of Companies in the Area	Pat Bailey	2019-03-7 05:28	6	Somewhat important
Video from Saturday	Laura DenHaan	2019-01-3 08:25	4	Somewhat important
Trending blogs right now	Tumblr	2019-03-1 07:39	0	Not important
How to use Trello at home	Taco from Trello	2019-01-24 10:23	0	Not important
50% OFF! Back to Business Sale Happening Now	Wix Team	2019-03-3 11:37	0	Not important

Exercise 9: Where the Data Comes From

The “Subject” input column will be pulled directly from the “Subject” line in the email. The “Sender” input column will be pulled directly from the “From” line in the email. The “Time Received” input column will be pulled directly from the timestamp on the email. The “Message response count from the past 2 weeks” input column will be pulled from a function that can go back through the messages from the last 2 weeks and count how many responses the user writes to each person. Then I applied the labels {very important, somewhat important, not important} to each email message that fell within a determined range of responses in the past 2 weeks.

Exercise 10: Easily Obtained Inputs

The easiest inputs to obtain would be the subject, sender, and time received since they would just be pulled directly from the email message themselves. However, the message response count from the past 2 weeks would be the most useful data since it is tailored to the user at that specific time and is based on some history. This is a harder data point to get since there needs to be some work done prior to getting the answer (going through each mail message in the last 2 weeks and counting how many times the user responded to each sender).