0.1 Question 1a

Generate your visualization in the cell below.

In [44]:
```
# Want to generate the proportion of ham/spam emails that contain the word.
# First, sum up the number of emails where the word is found, then divide by the total number
# np.sum along the index axis (sum the columns, not rows) because each column from the output
# are the occurrences for that word.
spam_words = ['href', 'free', 'act', 'congratulations', 'debt', 'income']

# Separate spam and ham emails
train_ham_emails = train.loc[train['spam'] == 0, 'email']
train_spam_emails = train.loc[train['spam'] == 1, 'email']

# Find number of ham emails containing the spam_words
words_in_ham = words_in_texts(spam_words, train_ham_emails)
summed_ham_words = np.sum(words_in_ham, axis=0)

# Find number of spam emails containing the spam_words
words_in_spam = words_in_texts(spam_words, train_spam_emails)
summed_spam_words = np.sum(words_in_spam, axis=0)

# Calculate the proportion of each group that contains the spam_words
prop_ham_words = summed_ham_words / len(train_ham_emails)
prop_spam_words = summed_spam_words / len(train_spam_emails)

# Create separate dfs and concat to create a full df
ham_df = pd.DataFrame(prop_ham_words.reshape(1, -1), columns=spam_words)
ham_df['type'] = 'Ham'
spam_df = pd.DataFrame(prop_spam_words.reshape(1, -1), columns=spam_words)
spam_df['type'] = 'Spam'
spam_ham_df = pd.concat([ham_df, spam_df]).melt('type')
spam_ham_df
```

Out[44]:

	type	variable	value
0	Ham	href	0.053441
1	Spam	href	0.508863
2	Ham	free	0.277212
3	Spam	free	0.491658
4	Ham	act	0.349777
5	Spam	act	0.549531
6	Ham	congratulations	0.002502
7	Spam	congratulations	0.021376
8	Ham	debt	0.005004
9	Spam	debt	0.072993
10	Ham	income	0.009115
11	Spam	income	0.088634

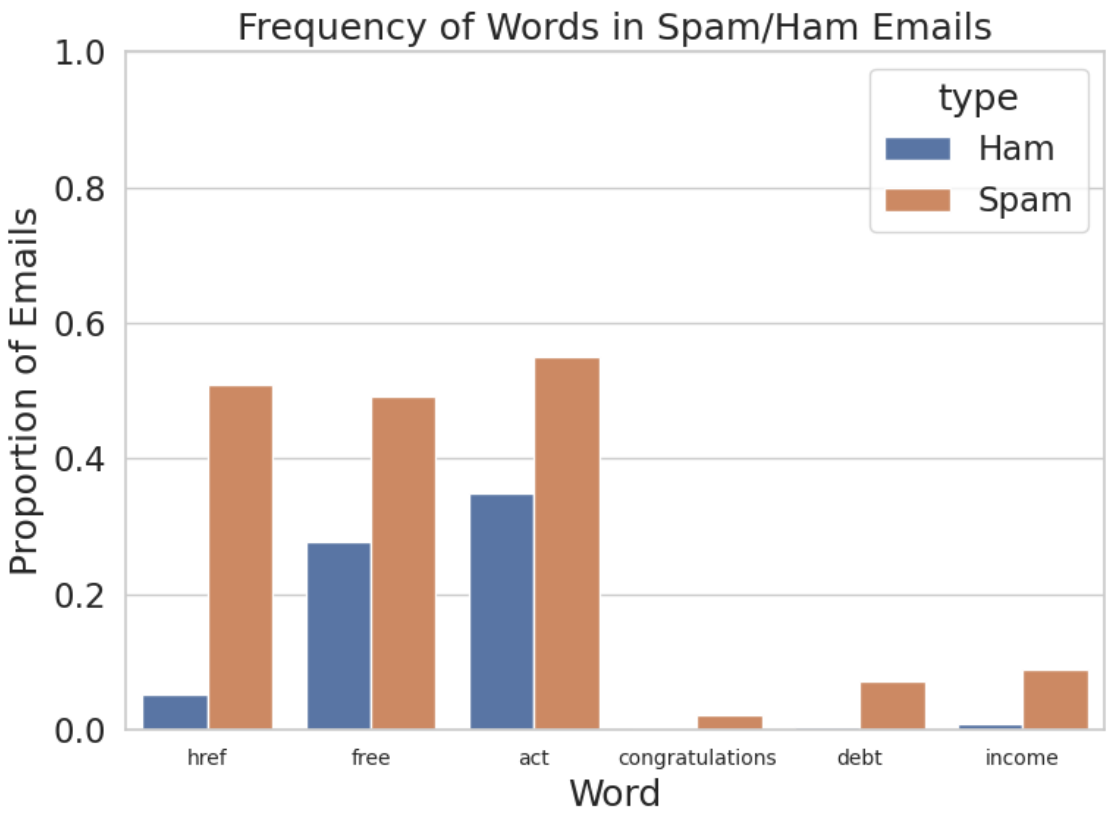
In [45]:
```
train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em
plt.figure(figsize=(8,6))
sns.barplot(data=spam_ham_df, x='variable', y='value', hue='type')

ax = plt.gca()
ax.set_ylim(0, 1)
ax.set_ylabel('Proportion of Emails')

ax.tick_params(axis='x', which='major', labelsize=10)
ax.set_xlabel('Word')

plt.title('Frequency of Words in Spam/Ham Emails')

plt.tight_layout()
plt.show()
```

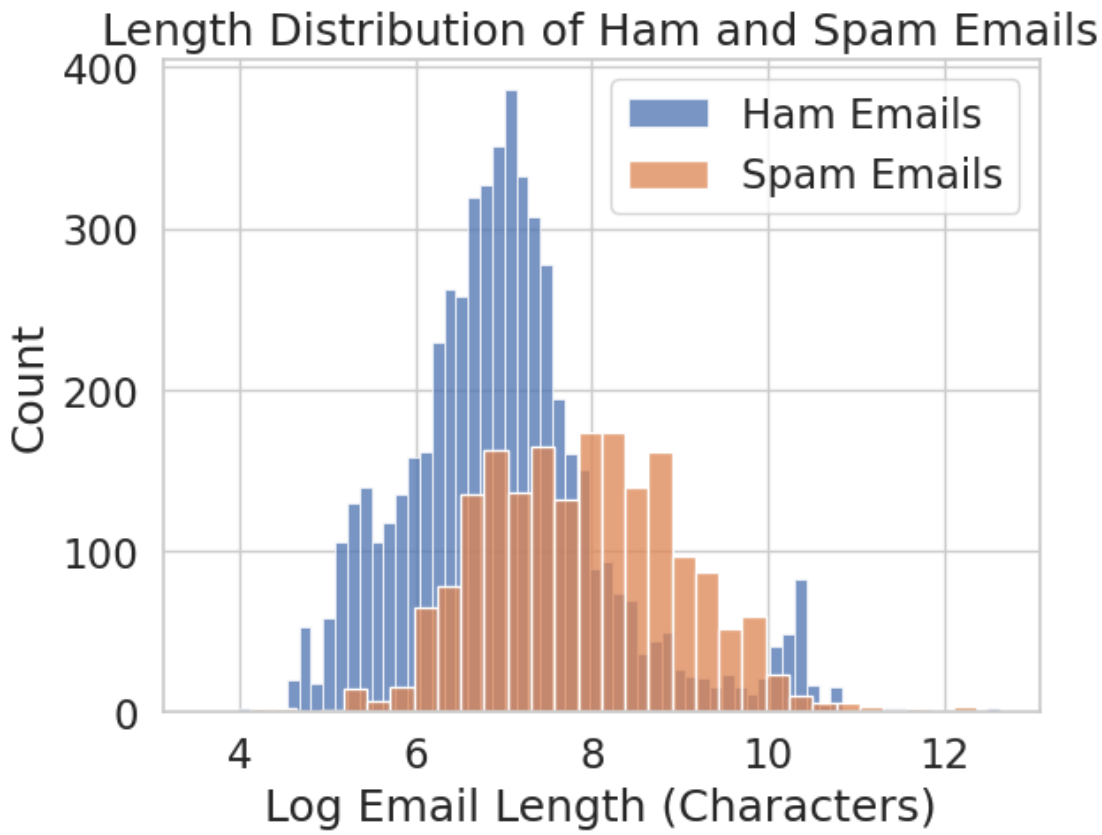


In [46]:
```
ham_email_lengths = train_ham_emails.str.len()
spam_email_lengths = train_spam_emails.str.len()

# ham_email_lengths.value_counts().sort_index()
```

```
# sns.histplot(ham_email_lengths.sort_values()[:-100], label='Ham Emails')
# sns.histplot(spam_email_lengths.sort_values()[:-20], label='Spam Emails')
sns.histplot(np.log(ham_email_lengths).sort_values(), label='Ham Emails')
sns.histplot(np.log(spam_email_lengths).sort_values(), label='Spam Emails')
plt.legend()
plt.xlabel('Log Email Length (Characters)')
plt.title('Length Distribution of Ham and Spam Emails')
```

```
Out[46]: Text(0.5, 1.0, 'Length Distribution of Ham and Spam Emails')
```




```
In [47]: train.head()
```

```
Out[47]:      id                                             subject  \
         0  7657                Subject: Patch to enable/disable log\n
         1  6911           Subject: When an engineer flaps his wings\n
         2  6074  Subject: Re: [Razor-users] razor plugins for m…
```

```
3  4376  Subject: NYTimes.com Article: Stop Those Press…
4  5766  Subject: What's facing FBI's new CIO? (Tech Up…
```

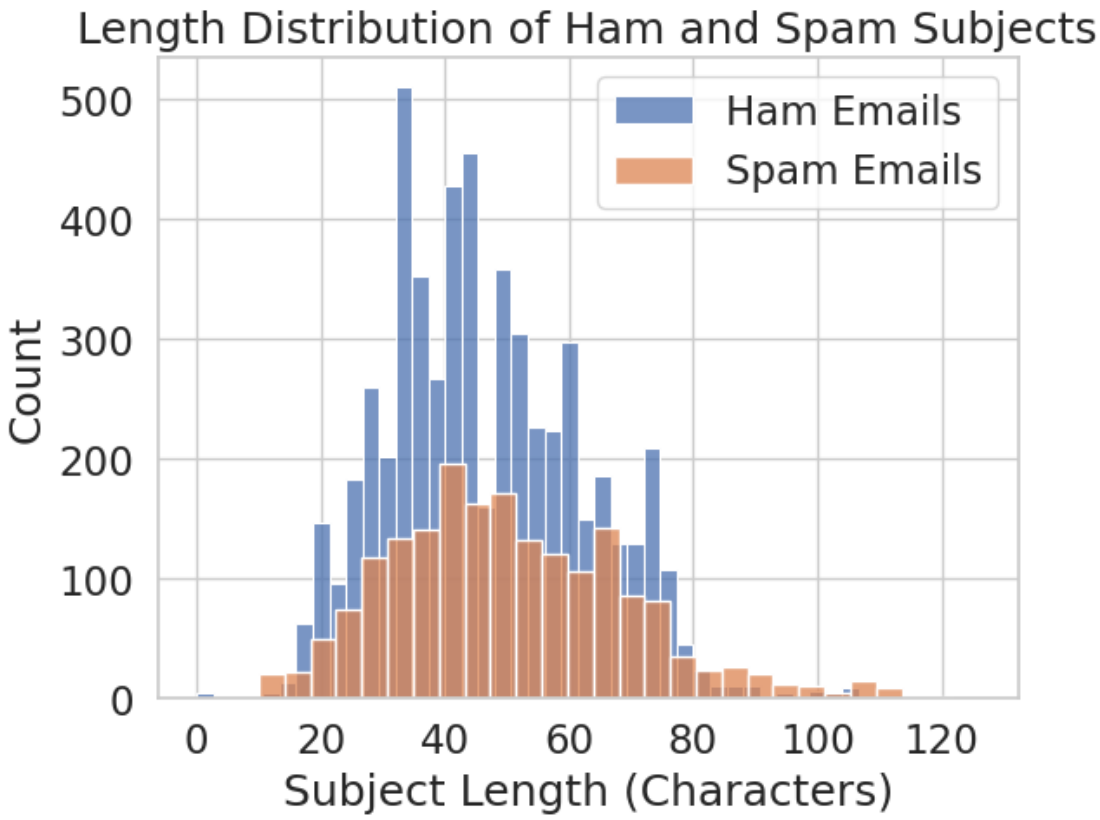
	email	spam
0	while i was playing with the past issues, it a…	0
1	url: http://diveintomark.org/archives/2002/10/…	0
2	no, please post a link!\n \n fox\n ----- origi…	0
3	this article from nytimes.com \n has been sent…	0
4	<html>\n <head>\n <title>tech update today</ti…	0

In [48]:
```
train_ham_subjects = train.loc[train['spam'] == 0, 'subject']
train_spam_subjects = train.loc[train['spam'] == 1, 'subject']

ham_subject_lengths = train_ham_subjects.str.len()
spam_subject_lengths = train_spam_subjects.str.len()

# ham_email_lengths.value_counts().sort_index()
# sns.histplot(ham_email_lengths.sort_values()[:-100], label='Ham Emails')
# sns.histplot(spam_email_lengths.sort_values()[:-20], label='Spam Emails')
sns.histplot(ham_subject_lengths.sort_values(), label='Ham Emails')
sns.histplot(spam_subject_lengths.sort_values(), label='Spam Emails')
plt.legend()
plt.xlabel('Subject Length (Characters)')
plt.title('Length Distribution of Ham and Spam Subjects')
```

Out[48]:
```
Text(0.5, 1.0, 'Length Distribution of Ham and Spam Subjects')
```



In [49]:
```
def is_reply(series):
    replies = ['Re:', 're:']
    return np.any(words_in_texts(replies, series), axis=1)
```

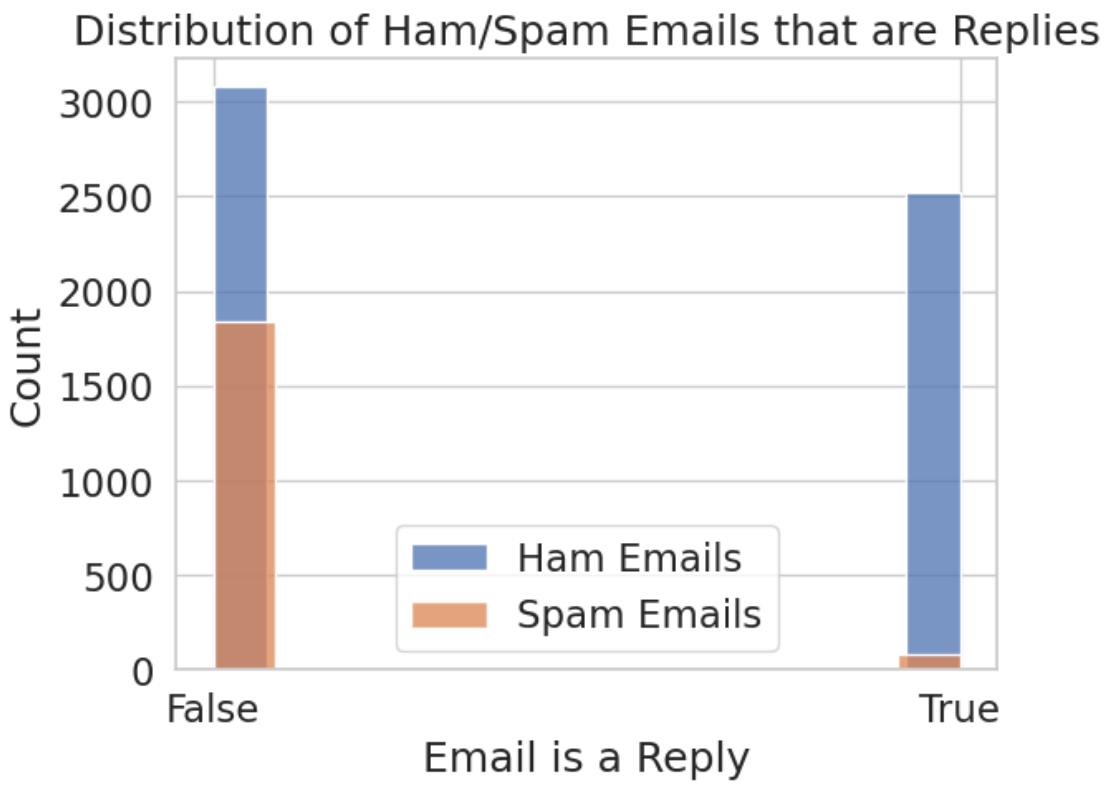
In [50]:
```
plt.clf()
ham_is_reply = is_reply(train_ham_subjects)
spam_is_reply = is_reply(train_spam_subjects)
display(ham_is_reply)

sns.histplot(ham_is_reply, label='Ham Emails')
sns.histplot(spam_is_reply, label='Spam Emails')
plt.legend()
plt.xlabel('Email is a Reply')
plt.xticks([0, 1], ['False', 'True'])
plt.title('Distribution of Ham/Spam Emails that are Replies')
```

```
array([False, False,  True, …, False,  True,  True])
```

```
<__array_function__ internals>:200: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'>
<__array_function__ internals>:200: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'>
```

Out[50]: Text(0.5, 1.0, 'Distribution of Ham/Spam Emails that are Replies')



0.2 Question 1b

In two to three sentences, describe what you plotted and its implications with respect to your features.

When plotting the distribution of email lengths of spam versus ham emails, it appears that the distribution is ever so slightly shifted rightwards for the spam emails versus the ham emails, suggesting that spam emails tend to be perhaps slightly larger in terms of length of characters compared to ham emails. When plotting the subject lengths, the distribution seems to be relatively similar. This perhaps indicates that both subject and email length may not be very indicative of ham or spam emails.

Interestingly, when plotting the distribution of Ham/Spam emails that are replies, we see a huge shift. The vast majority of spam emails are NOT replies, whereas a large proportion of Ham emails are! When compared to the differences in length distributions between the two samples, it appears clear from plotting that the "Is a Reply" feature has more of a clear distinction between spam and ham emails. Thus, I will prioritize including the "Is a Reply" feature in my model over length

1 Question 4

Describe the process of improving your model. You should use at least 2-3 sentences each to address the following questions:

1. How did you find better features for your model?
2. What did you try that worked or didn't work?
3. What was surprising in your search for good features?

1. I found new feature mainly through EDA, plotting various features and seeing if their distributions varied between the spam and ham emails. For example, plotting the distribution of certain words and seeing if they had different proportions of appearances between spam and ham emails allowed me to determine if I should include those words in my word list to pass into the words_in_texts feature function. Conversely, when I plotted the distribution of spam/ham emails that are replies (found by finding if the subject line contains Re: or re:), I saw a very large difference between the two samples. Thus, I followed this logic and EDA to determine features that are more closely correlated with spam/ham emails for my model.
2. I tried performing a simple count of all characters in the subject / email body, giving a rough estimate of email length. However, when first plotting these features, there are a few outliers in both spam/ham emails where the email is extremely long, skewing the data to have a long right-tail (right-skewed). Thus I tried performing a log transformation on email body length, in number of characters, and feeding that into the model. However, when checking the coefficients of my logistic regression model, the absolute value of the weight corresponding to email body length was close to 0, meaning the feature was very minorly impactful on classifying spam/ham if it was helpful at all. Thus, the length of emails didn't seem to work as a feature. Conversely, the "Reply" feature seemed to work very well, having a strongly negative weight of greater than 2, meaning it was quite impactful on determining if an email was spam or ham.
3. It was surprising that length of the body of the email doesn't really correlate strongly with spam or ham emails; I would've thought that spam emails may be longer as they contain hyperlinks or other random garbage, but it looks like quite a few ham emails are also very long. Additionally, it was surprising that certain words which don't appear super frequently in the total population of emails, such as "Debt" or "Income", still have a relatively high weights in our model. This means that, although these words don't appear very frequently in the whole sample, when they do appear, they are usually quite indicative of the email being spam.

2 Question 5: ROC Curve

In most cases, we won't be able to get 0 false positives and 0 false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover that they have cancer until it's too late. In contrast, a patient can receive another screening for a false positive.

Recall that logistic regression calculates the probability that an example belongs to a particular class. To classify an example, we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, **we can adjust that cutoff threshold**. We can say that an email is spam only if our classifier gives it ≥ 0.7 probability of being spam, for example. This is how we can trade off false positives and false negatives.

The Receiver Operating Characteristic (ROC) curve shows this trade-off for each possible cutoff probability. In the cell below, plot an ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Lecture 23 may be helpful.

Hint: You'll want to use the `.predict_proba` method (documentation) for your classifier instead of `.predict` to get probabilities instead of binary predictions.

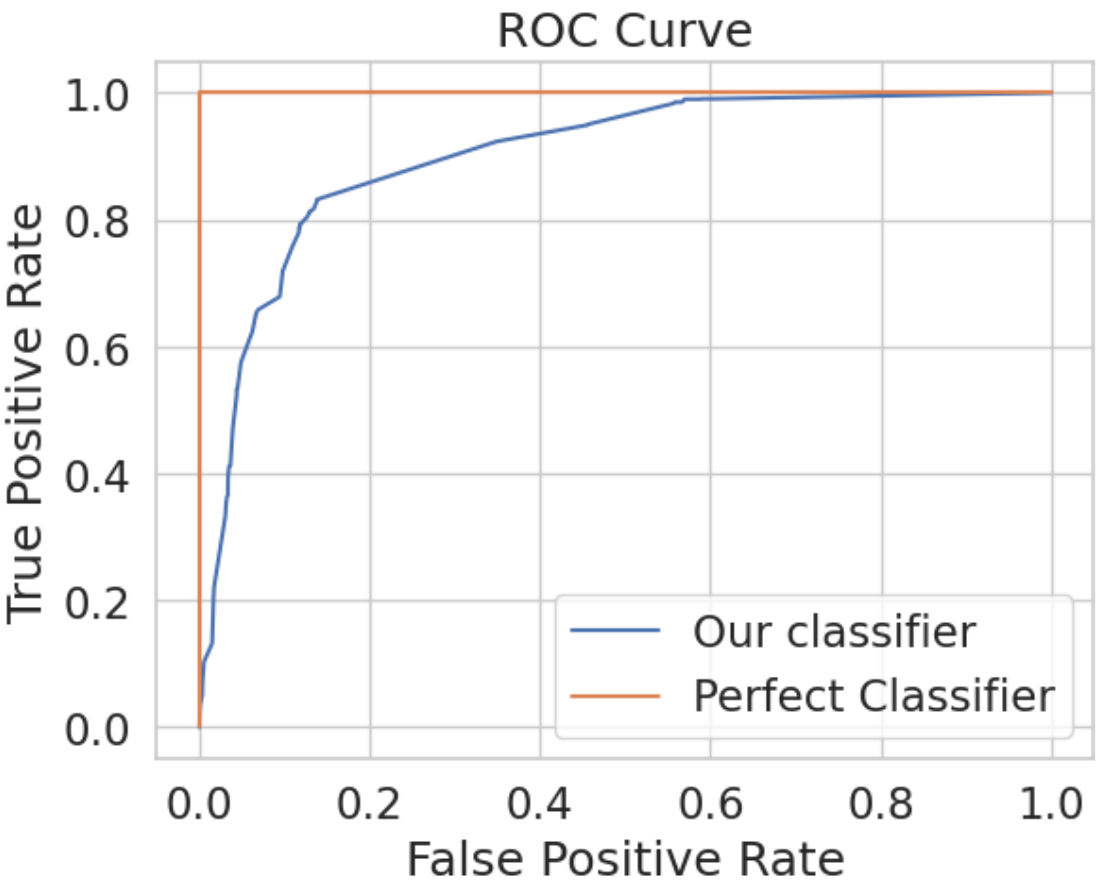
```
In [67]: def predict_threshold(model, X, T):
             prob_one = model.predict_proba(X)[:, 1]
             return (prob_one >= T).astype(int)

         def tpr_threshold(X, Y, T): # Same as recall
             Y_hat = predict_threshold(my_model, X, T)
             return np.sum((Y_hat == 1) & (Y == 1)) / np.sum(Y == 1)

         def fpr_threshold(X, Y, T):
             Y_hat = predict_threshold(my_model, X, T)
             return np.sum((Y_hat == 1) & (Y == 0)) / np.sum(Y == 0)

         thresholds = np.linspace(0, 1, 100)
         tprs = [tpr_threshold(train_features, train_labels, t) for t in thresholds]
         fprs = [fpr_threshold(train_features, train_labels, t) for t in thresholds]
```

```
In [68]: plt.plot(fprs, tprs, label="Our classifier")
         plt.plot([0, 0, 1], [0, 1, 1], label="Perfect Classifier")
         plt.xlabel("False Positive Rate")
         plt.ylabel("True Positive Rate")
         plt.title('ROC Curve')
         plt.legend();
```


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
Our classifier
Perfect Classifier

2.0.1 Question 6a

Pick at least **one** of the emails provided above to comment on. How would you classify the email (e.g., spam or ham), and does this align with the classification provided in the training data? What could be a reason someone would disagree with *your* classification of the email? In 2-3 sentences, explain your perspective and potential reasons for disagreement.

For example 1, the email is classified as spam in the training data, but I personally believe that it is a regular, non-spam email. The reason I believe this email is not a spam email is because the body of the email seems to be genuinely written by an individual who wants to connect with her family or loved ones, and it contains some reasonable spelling errors that are human-like. However, someone may disagree with this classification because the email is sent as a reply, and contains the original email being replied to in the body of the text. In this original email, there are the words "travelogue mailing list," presumably from the person's signature (perhaps he sent a personal email from his business email or something). Thus, the signature may seem like the email is a spam email, but when reading the actual context of the emails, it appears to me that the email is not spam.

2.0.2 Question 6b

As data scientists, we sometimes take the data to be a fixed "ground truth," establishing the "correct" classification of emails. However, as you might have seen above, some emails can be ambiguous; people may disagree about whether an email is actually spam or ham. How does the ambiguity in our labeled data (spam or ham) affect our understanding of the model's predictions and the way we measure/evaluate our model's performance?

Obviously, if the definition of what spam or ham emails are is subjective, this changes how we evaluate our model performance because the data and labels we train on may not necessarily be "accurate," or we may have to start questioning the ground truth of our training data. Thus, it becomes more difficult to just trust accuracy of our model in predicting labels because the training labels may not be accurate themselves. Additionally, this may make it more difficult to generalize our model; if we were to generalize it, it would need to be used on data that was acquired with the exact same criteria and labeling rationale as this original training data, to try and account for the subjectivity of spam/ham emails from different people.

Part ii In 2-3 sentences, explain why you think the feature you chose to remove changed how your email was classified.

I chose an email that didn't contain any of the other words in the some_words list, but only contained the word 'bank'. Therefore, by removing the word 'bank' from the some_words list, and thus removing it as a feature, the model no longer had that variable to distinguish the email as spam. Additionally, because the weight corresponding to the 'bank' word feature is relatively high (1.413033), removing this feature has a relatively high impact on the model prediction.

Part i In this context, do you think you could easily find a feature that could change an email's classification as you did in part a)? Why or why not?

No I don't believe so. This is because, with a model with many more features, the removal of one single feature is less likely to make a significant impact on the prediction. Especially with regularization techniques in place, a model with multiple features is more likely to have weights of smaller values, meaning each individual feature matters less to the overall prediction. Therefore, it would be very difficult to find a single feature to delete that would be weighted so heavily that it could change an email's classification.

Part ii Would you expect this new model to be more or less interpretable than `simple_model`?

Note: A model is considered interpretable if you can easily understand the reasoning behind its predictions and classifications. For example, the model we saw in part a), `simple_model`, is considered interpretable as we can identify which features contribute to an email's classification.

This model would be less interpretable than the simple model because there are significantly more features. It would become less clear which features are extremely important for the model, and which features are less important individually and only contribute holistically to the model. The more features there are, the less interpretable the model will become because it is difficult to understand what exactly each feature is contributing.

2.0.3 Question 7c

Now, imagine you're a data scientist at Meta, developing a text classification model to decide whether to remove certain posts / comments on Facebook. In particular, you're primarily working on moderating the following categories of content: * Hate speech * Misinformation * Violence and incitement

Pick one of these types of content to focus on (or if you have another type you'd like to focus on, feel free to comment on that!). What content would fall under the category you've chosen? Refer to Facebook's Community Standards, which outline what is and isn't allowed on Facebook.

If I were to focus on Hate Speech, the type of content that would fall under this category would include any pointed posts or videos that encourage hatred and violence against a specific group of people, solely based on their religion, race, sex, etc. This kind of content would not be allowed under the Community Standards, as not only does it violate the dignity of individuals within the targeted group, but also threatens the safety of members of that group. Social media that encourages hatred towards a particular group could have long-standing and widespread social consequences; for example, increased random acts of violence against Asian people after increased hate speech against China occurred due to the outbreak of coronavirus.

2.0.4 Question 7d

What are the stakes of misclassifying a post in the context of a social media platform? Comment on what a false positive or false negative means for the category of content you've chosen (hate speech, misinformation, or violence and incitement).

The mistake of misclassifying a post could lead to an infringement of people's voices, and an infringment of the "authenticity" of Facebook. Overzealous removal of posts could almost be akin to censorship, where any quote on quote "negative posts" could just be removed, leading to a deafening of a voices for a certain group of people with the same ideology. There is a fine balance between allowing people to excercise their right to their own opinion and voice, while also maintaining public safety and ensuring those voices aren't being used to encourage violence against another group. A False Positive for hate speech would be someone mentioning a particular gender, race, religion, or group of people in their post but not saying anything hateful at all, just simply mentioning them in their post, but being flagged for hate speech. Another good, relevant example for today's political climate would be posts referencing "Free Palestine," which obviously refer to Palestinians but if falsely classified as hate speech would be removed and thus remove a good number of voices supporting Palestine and ceasefire. A false negative for hate speech would be a post that is overtly racist or hateful towards a group, but isn't detected by the model. An example could be a post referencing a racial slur, or insinuating hatred against a group of people in more subtle terms, but not overtly typing any race or group of people in the post.

2.0.5 Question 7e

As a data scientist, why might having an interpretable model be useful when moderating content online?

An interpretable model may be more useful for moderating content because it is easier to understand how to fine-tune the model to better detect hate speech, or reduce False Positives and reduce False Negatives. For example, if features are simpler such as simple words, one can always add another feature corresponding to a new hate word that the model can detect. If the model is more complex and less interpretable, it becomes difficult to perform fine-tuning of the model.