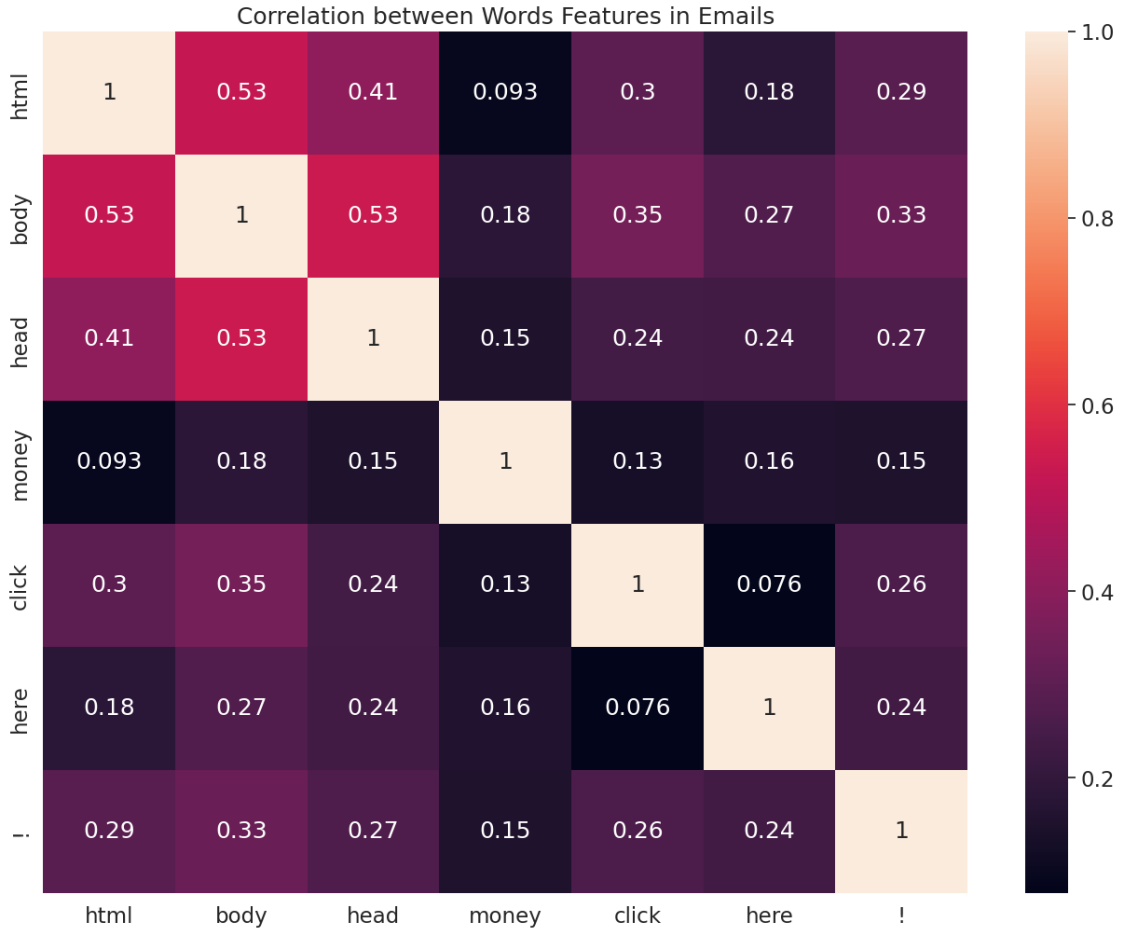
0.1 Question 1a

Generate your visualization in the cell below.

```
In [40]: words = ['html', 'body', 'head', 'money', 'click', 'here', '!']

         yes_words = words_in_texts(words, train['email'])
         words_df = pd.DataFrame(yes_words, columns=words)

         plt.figure(figsize=(16, 12))
         sns.heatmap(words_df.corr(), annot=True)
         plt.title("Correlation between Words Features in Emails");
```



0.2 Question 1b

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

I plotted a heatmap to show the correlation between selected words, where each word was represented as a binary feature (1 if present in an email, 0 otherwise). By examining how often two words occurred together, I wanted to identify redundant features and overlapping patterns that might affect model performance. I moreso focused on words commonly found in spam emails (but can still be seen in ham emails) to better understand which combinations reflect similar content.

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 better features for my model by expanding my word list and selecting terms I thought could help distinguish spam from ham emails. I used the heatmap in Q1a to observe how strongly each word correlated with each other to see if they would exist in the same email, which helped me prioritize features that showed higher predictive potential. This visualization allowed me to eliminate words that had weak associations and focus on those with clearer patterns.
2. I began with a broad set of words I thought could distinguish between spam and ham emails. Although my code doesn't show how often each word appears in either category, I noticed signs of overfitting. To address this, I refined my feature list using intuition and feedback from model performance. Many of the initial words were too general and didn't improve accuracy, so I narrowed the list to more specific, spam-related terms. This adjustment helped improve the model's training accuracy.
3. One thing that surprised me was that including more spam-like words like "click," "money," or "win" didn't improve the model's accuracy as much as I expected. Since my code doesn't distinguish between spam and ham emails, this suggests that these words might also appear frequently in ham messages. This made me realize that simply focusing on individual words may not be the most effective strategy and that considering more targeted features or combinations of words might yield better results.

2 Question 5: ROC Curve

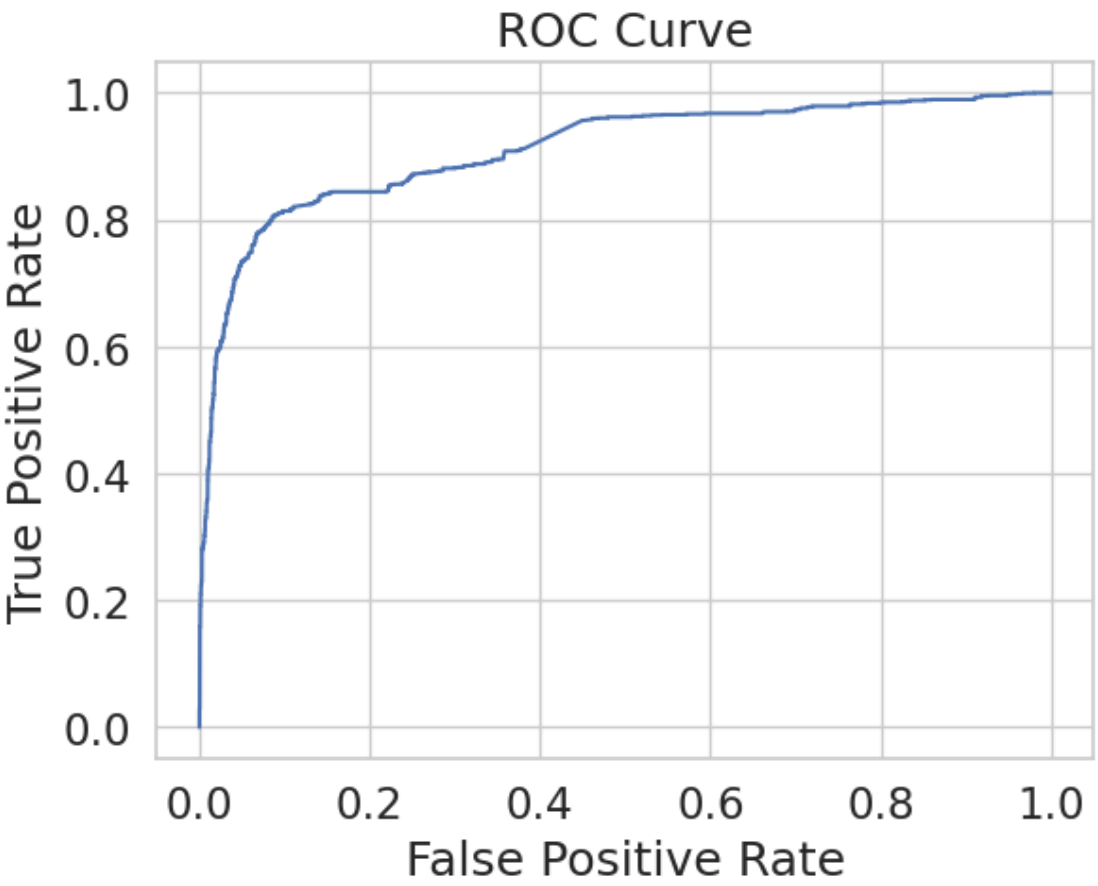
In most cases, we won't be able to get 0 false positives and 0 false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover that they have cancer until 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 [49]: Y_predict = model.predict_proba(X_train)[:,1]
         fpr, tpr, thresholds = roc_curve(Y_train, Y_predict)
         plt.plot(fpr, tpr)
         plt.title('ROC Curve');
         plt.xlabel('False Positive Rate');
         plt.ylabel('True Positive Rate');
```


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0

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.

I would classify the first email as ham, even though it is labeled as spam in the training data. The message reads like a personal message and reply to a friend named 'Justin Mason', which contains content about an upcoming meetup and celebrating the birth of a child. The email's content is conversational and specific to the individuals, which aligns more with ham content. Someone might disagree with my classification because the email contains informal language, such as 'vv busybeing' and excessive punctuations like '!!!', which might trigger spam filters.

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?

Ambiguity in our labeled data makes it harder to trust whether classifications are truly "correct". As seen in Q6a, I disagreed with a provided label, showing that what counts as "spam" or "ham" can be subjective. If labels are inconsistent or debatable, our model might be penalized for making reasonable predictions. This uncertainty affects how we evaluate the model. Metrics like accuracy, precision, or recall become less reliable because they assume a fixed "ground truth" that may not exist for ambiguous cases.

Part ii Please provide below the index of the email that you flipped classes (`email_idx`). Additionally, in 2-3 sentences, explain why you think the feature you chose to remove changed how your email was classified.

The index of the email I flipped was `email_idx` $ = 27$. In part i, I filtered through the emails in the training set and found an email that contained one of the five possible features. By removing that feature, it changed the classification because the presence of that word was strongly associated with spam, and removing it made the email appear less like spam to the model. Without that feature, the email's content changed, which caused the model to re-evaluate its classification decision.

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?

I think I could not easily find a feature that could change an email's classification as I did in part a because in a larger model with many features, the model relies on the interaction of many features, and no single feature has as much weight individually. Identifying and removing a single feature would likely have less of an impact on the classification compared to a smaller model, where a few features might have a more significant influence. The model's decision depends on the combined effect of many features, so removing one feature may not cause a noticeable change in the 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.

As mentioned above, I would expect this new model to be less interpretable than `simple_model`. As the number of features increases, it becomes harder to understand each feature's contribution to the model's predictions. In `simple_model`, which uses a small set of features, it is easy to see how each feature impacts the classification (ie. whether 'drug' or 'bank' can be found in the email). However, in a model with significantly more features, the interactions between features become more complex, making it difficult to figure our which features are most influential or how they combine to produce the final prediction. This leads to a decrease in interpretability because the reasoning behind the model's classification becomes less transparent.

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.

Content that falls under "Violence and Incitement" includes threats directed at individuals or specific groups, promotion or encouragement of self-harm or suicide, verbal or physical abuse, and posts that attempt to organize or promote criminal activity. It also includes content that glorifies past violence in ways that could inspire copycat behavior, encourages others to commit acts of violence or crime, or promotes hate-based events that have the potential to result in real-life harm or injuries.

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 and false negative means for the category of content you've chosen (hate speech, misinformation, or violence and incitement).

The stakes of misclassifying a post in the context of a social media platform are high, especially for content related to violence and incitement. Misclassifications can expose users, like those with trauma or sensitivity to violent content, to harmful material or unfairly censor individuals who haven't violated community standards. A false positive occurs when a post is mistakenly flagged as violent or inciting harm when it isn't. While this might not cause physical harm, it can result in unnecessary content removal and lose trust in the platform's moderation. However, a false negative allows harmful content to remain online, which can lead to consequences, such as triggering trauma, encouraging violent behavior, or facilitating criminal activity. This is a more serious risk than a false positive.

2.0.5 Question 7e

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

Having an interpretable model is useful when moderating content online because it makes it easier to understand what was flagged and why. This helps data scientists evaluate how well the model is performing, identify sources of error, and improve moderation accuracy. It also supports transparency and helps ensure that decisions align with community guidelines and standards.