0.1 Question 1a

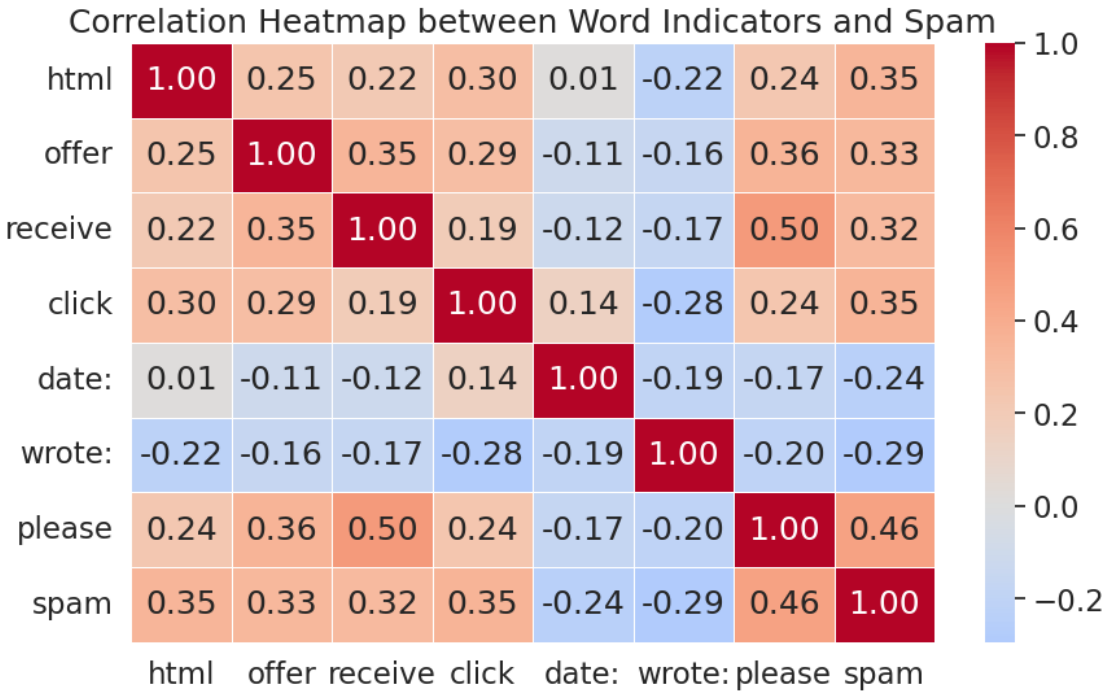
Generate your visualization in the cell below.

```
In [12]: words = ['html', 'offer','receive','click','date:','wrote:','please']

         by_word = pd.DataFrame(words_in_texts(words, train['email']))
         by_word = by_word.rename(columns=dict([(k,v) for k,v in zip([0,1,2,3,4,5,6], words)]))
         by_word['spam'] = train['spam']

         correlation_matrix = by_word.corr()

         plt.figure(figsize=(10, 6))
         sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", center=0, fmt=".2f", linewidths=0
         plt.title('Correlation Heatmap between Word Indicators and Spam')
         plt.show()
```



0.2 Question 1b

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

The heatmap visualizes the correlation between certain keywords and their association with spam emails. The strong positive correlation between words like "click" and "html" with the spam label suggests that these terms are common in spam emails, making them strong candidates for predictive features in spam detection models. Conversely, the negative correlations between words like "wrote:" with others indicate that it may carry different weight in the context of spam, potentially reducing its predictive value when it appears with certain terms. This kind of analysis can guide the feature selection process by highlighting which terms are most characteristic of spam or ham emails, thereby informing a more effective machine learning model.

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?

To find better features for my model, I started by looking at the text of emails to find words that more strongly associate with ham/spam emails and performed feature extractions, such as the number of words in the subject and email and the number of special characters. These features were inspired by the assumption that spam emails may have distinct characteristics in terms of length or punctuation usage. I used the word indicator (words_in_texts) based on prior knowledge of common terms in spam emails. What worked well was incorporating word indicator features using the words_in_texts function, which enabled the model to better differentiate between spam and non-spam emails based on specific terms. Adding too many words as features made the model too complex. Also, what didn't work as well was relying solely on basic length-based features (like word count and special character count), as they didn't provide enough discriminative power between spam and ham emails, and the model's recall for spam emails remained low when using only these features. One surprising finding was that word indicators (specific words found in spam emails) provided more meaningful features than I initially expected. Words like "please", "html", and "click" showed strong correlations with spam emails, which helped the model detect spam better. On the other hand, simpler features like the length of the subject or body of the email didn't have as strong a correlation with spam as anticipated, and they didn't help improve the model's performance much when used alone. This highlighted the importance of focusing on more semantically rich features, like specific words or phrases, rather than just structural features.

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.

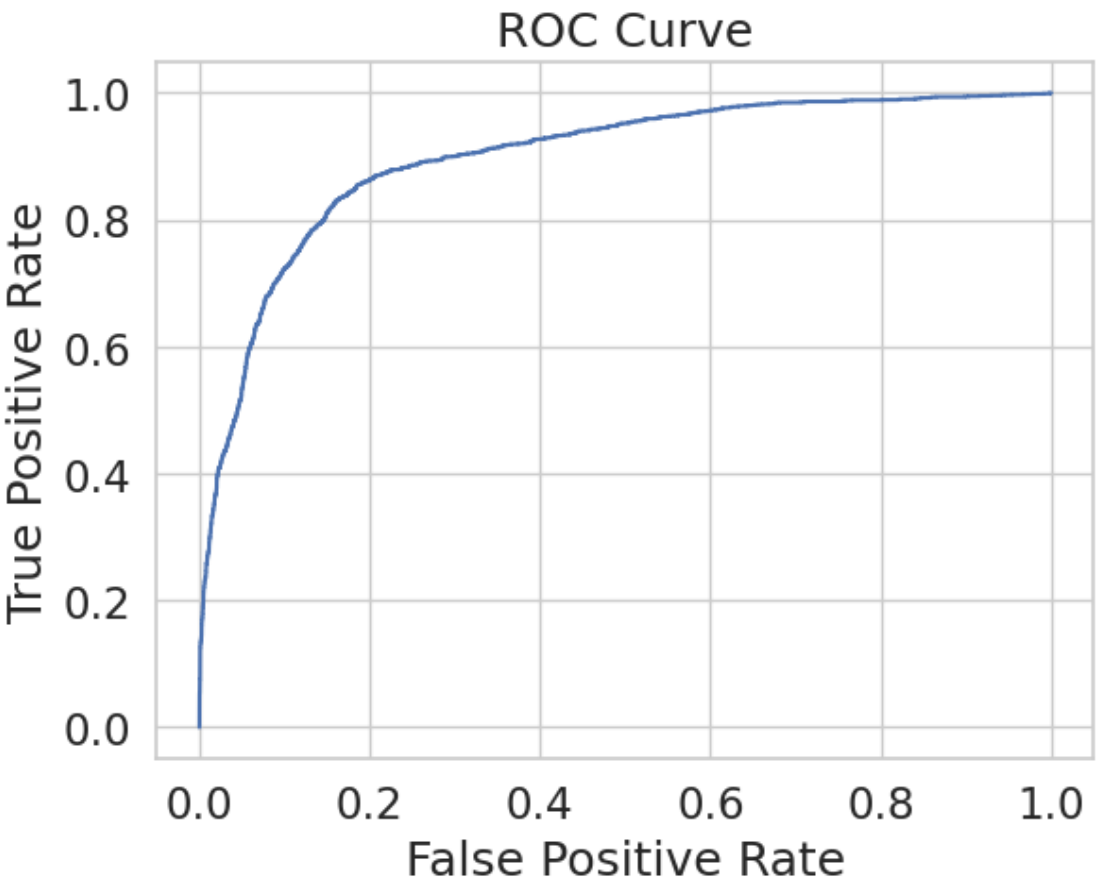
Recall that logistic regression calculates the probability that an example belongs to a particular class. To classify an example, we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, **we can adjust that cutoff threshold**. We can say that an email is spam 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 [21]: probability = model.predict_proba(X_train)[:,1]
         FPR, TPR, thresholds = roc_curve(Y_train, probability, pos_label=1)
         plt.plot(FPR, TPR)
         plt.title("ROC Curve")
         plt.xlabel('False Positive Rate')
         plt.ylabel('True Positive Rate')
```

```
Out[21]: Text(0, 0.5, '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.

This email would likely be classified as ham. This aligns with the classification in the training data because the content of the email is a newsletter from Lifetime TV, promoting their upcoming movies and TV events. One might argue that the email could be classified as spam because it is unsolicited, promoting a commercial product (Lifetime TV movies), and contains multiple links to external websites. However, from the content and structure of the email, it appears to be a legitimate promotional email from a known company, which typically wouldn't be classified as spam in a traditional sense because it doesn't exhibit characteristics commonly associated with spam, such as aggressive sales language or a sense of urgency.

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 labels can create noisy training data, where the model learns from incorrect or inconsistent labels. If some emails that are labeled as spam are actually more like ham, and vice versa, the model may overfit to these inconsistencies, resulting in poor generalization to unseen data. When there's disagreement in the labeling of data, it becomes harder to assess model robustness and confidence, and evaluations (accuracy and precision/recall) become misleading.

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.

I chose email index 50. It was originally classified as spam (probability of 55.57%), but now would be classified as ham (24.33% probability of being classified spam). The feature 'bank' was likely a strong indicator of the email being spam, as emails containing financial terms like "bank" are often associated with spam emails (e.g., phishing attempts, unsolicited offers). Removing this feature caused the model to re-evaluate the email, leading to a lower probability of being spam (from 55.57% to 24.33%).

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?

With 1000 features, the number of possible interactions and dependencies between features increases exponentially. This makes it more difficult to isolate one individual feature whose removal will lead to a significant change in classification. In a model with fewer features, each feature has a more direct influence on the classification. However, in larger models, many features are likely correlated or interdependent, so removing one feature might have a smaller or less noticeable impact, especially if the model is more robust. Also, the weights of features in larger models may be less significant than for features in smaller models, so the impact a single feature makes in a large model will have little impact on the output.

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 new, lager model would be less interpretable because it is harder to track individual feature combinations to the model's predictions. Even though the model may produce more accurate results, the sheer number of features introduces complexity and interactions that are difficult to interpret. You might not know which exact features are most influential in a given classification. In smaller models, the relationship between features and the target variable is usually simpler and more direct. You can easily tell which feature caused a particular prediction or classification (e.g., "bank" indicating a financial scam email).

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.

Hate speech refers to any form of expression, whether verbal, written, or visual, that incites hate, violence, or discrimination against individuals or groups based on attributes such as race, ethnicity, nationality, religion, gender, sexual orientation, disability, or other protected characteristics. According to Facebook's Community Standards, hate speech is prohibited because it can contribute to harmful or dangerous behavior. Posts that use derogatory terms or insults targeting individuals or groups based on their race or ethnicity would be considered hate speech. For example, slurs against Black people or any ethnic group. Likewise, content that encourages violence or harm against a specific group or individual. For instance, a post that promotes violence against a particular religious group or ethnic minority.

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 are high because misclassifications can impact user safety, the platform's credibility, and overall public trust. A false positive occurs when a post that does not contain hate speech is wrongly flagged as hate speech. One of the key risks of false positives is that legitimate expression could be wrongly suppressed, suppressing a user's freedom of speech. This could result in user disengagement and would negatively impact the reputation of the company. A false negative occurs when a post that does contain hate speech is not flagged by the system, allowing harmful content to remain visible. Hate speech that goes unaddressed can contribute to real-world harm, including violence, discrimination, and harassment. Allowing hate speech to persist on the platform can lead to public criticism and reputational damage.

2.0.5 Question 7e

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

Accountability is essential for platforms moderating content, as users should have clarity on why certain posts are flagged or removed. If users are unsure about the decision-making process, it can lead to frustration, mistrust, and a lack of confidence in the moderation system. Furthermore, an interpretable model enables data scientists to understand why a specific classification (e.g., spam, hate speech) was made. This is essential for identifying errors and biases in the model. For example, if the model consistently flags certain harmless posts as hate speech, understanding why can guide model refinements and reduce biases and harm.