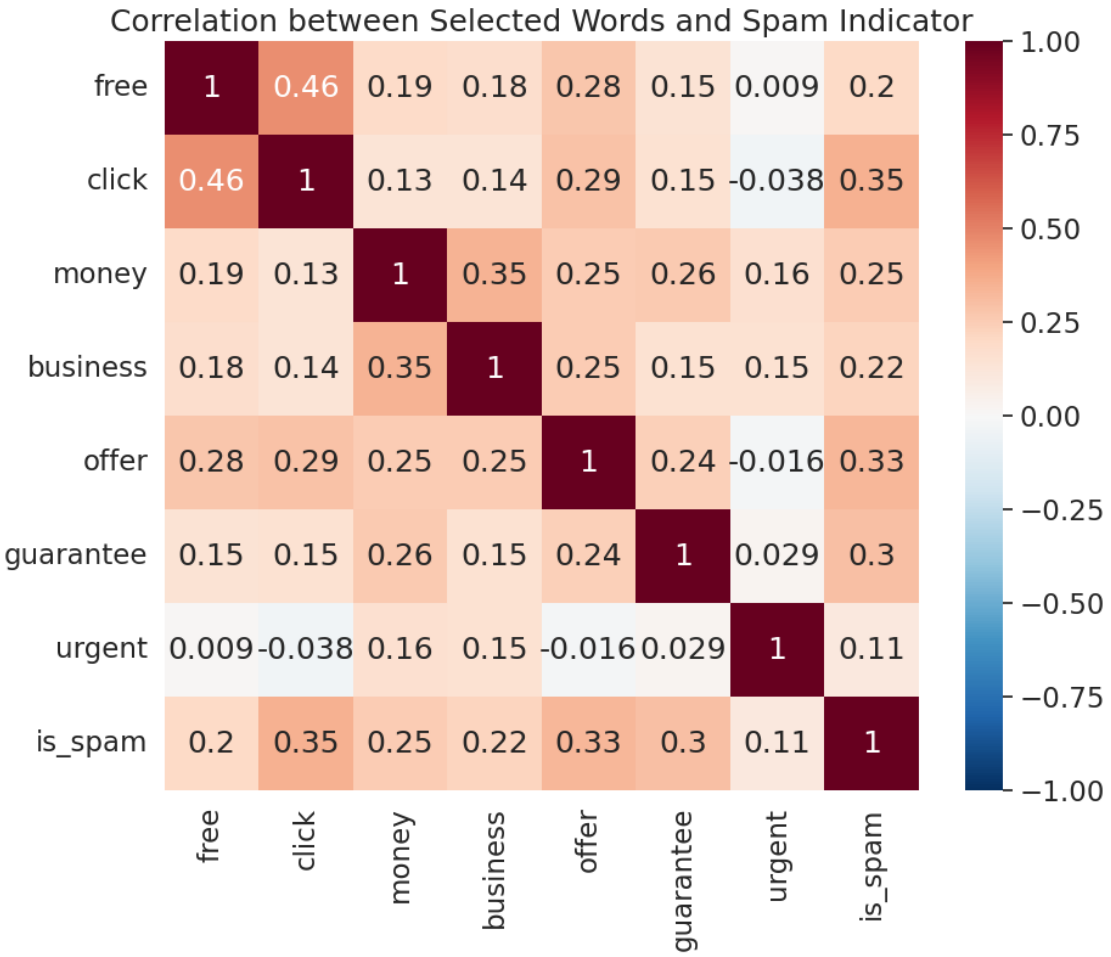
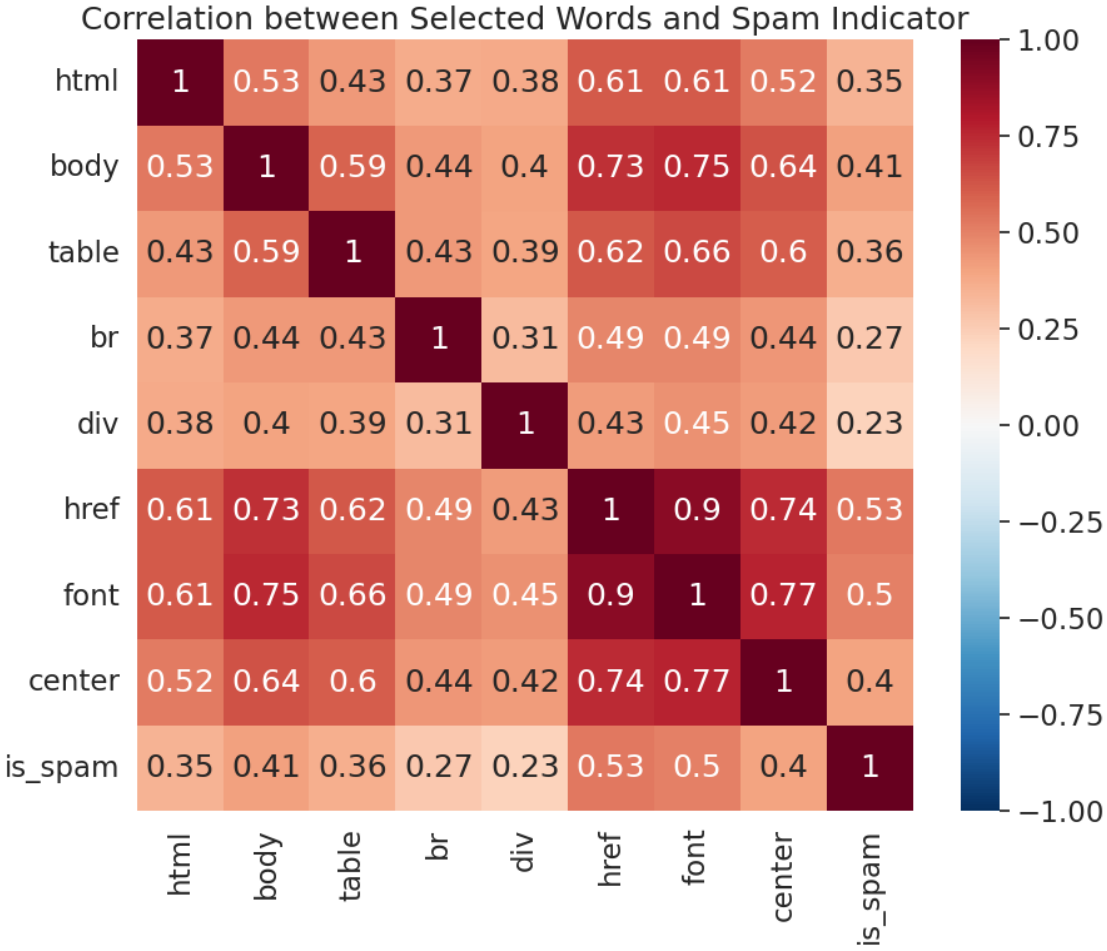

0.1 Question 1a

Generate your visualization in the cell below. As a friendly reminder, choose some plot other than the 1-dimensional distribution of some quantity for spam and ham emails.

In [14]:
```
import seaborn as sns
import matplotlib.pyplot as plt

words = ['free', 'click', 'money', 'business', 'offer', 'guarantee', 'urgent']

indicator_matrix = words_in_texts(words, train['email'])
df_features = pd.DataFrame(indicator_matrix, columns=words)
df_features['is_spam'] = train['spam'].reset_index(drop=True)
corr_matrix = df_features.corr()

plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='RdBu_r', vmin=-1, vmax=1)
plt.title('Correlation between Selected Words and Spam Indicator')
plt.show()
```




```
In [15]: words2 = ['html', 'body', 'table', 'br', 'div', 'href', 'font', 'center']

         indicator_matrix2 = words_in_texts(words2, train['email'])
         df_features2 = pd.DataFrame(indicator_matrix2, columns=words2)
         df_features2['is_spam'] = train['spam'].reset_index(drop=True)
         corr_matrix2 = df_features2.corr()

         plt.figure(figsize=(10, 8))
         sns.heatmap(corr_matrix2, annot=True, cmap='RdBu_r', vmin=-1, vmax=1)
         plt.title('Correlation between Selected Words and Spam Indicator')
         plt.show()
```


Correlation between Selected Words and Spam Indicator
html 1 0.53 0.43 0.37 0.38 0.61 0.61 0.52 0.35
body 0.53 1 0.59 0.44 0.4 0.73 0.75 0.64 0.41
table 0.43 0.59 1 0.43 0.39 0.62 0.66 0.6 0.36
br 0.37 0.44 0.43 1 0.31 0.49 0.49 0.44 0.27
div 0.38 0.4 0.39 0.31 1 0.43 0.45 0.42 0.23
href 0.61 0.73 0.62 0.49 0.43 1 0.9 0.74 0.53
font 0.61 0.75 0.66 0.49 0.45 0.9 1 0.77 0.5
center 0.52 0.64 0.6 0.44 0.42 0.74 0.77 1 0.4
is_spam 0.35 0.41 0.36 0.27 0.23 0.53 0.5 0.4 1
html
body
table
br
div
href
font
center
is_spam
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

0.2 Question 1b

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

The heatmap I have created shows the relationship between selected word features and the is_spam label, to check for redundancy between the features. The plot (second heatmap) reveals that structural HTML tags like font, href, and body have relatively strong positive correlations (dark red colored squares) with is_spam, confirming that the presence of heavy HTML formatting is a strong indicator of spam compared to plain text emails.

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?

I first relied on a small, simple list of obvious words (such as "money" or "free"), which did not work well, as it failed to catch more subtle types of spam. However, then I decided to raise the number of word features, especially HTML tags, that seemed to work better as it improved my accuracy. This suprised me as I didn't expect features like HTML tags to be stronger predictors of spam than words like "win" or "cash." I also decided to refine the model's optimization by switching to the liblinear solver, which proved stable for this dataset, increased max_iter to 1000 to ensure full convergence, and used the l2 regularization to prevent overfitting.

2 Question 5: ROC Curve

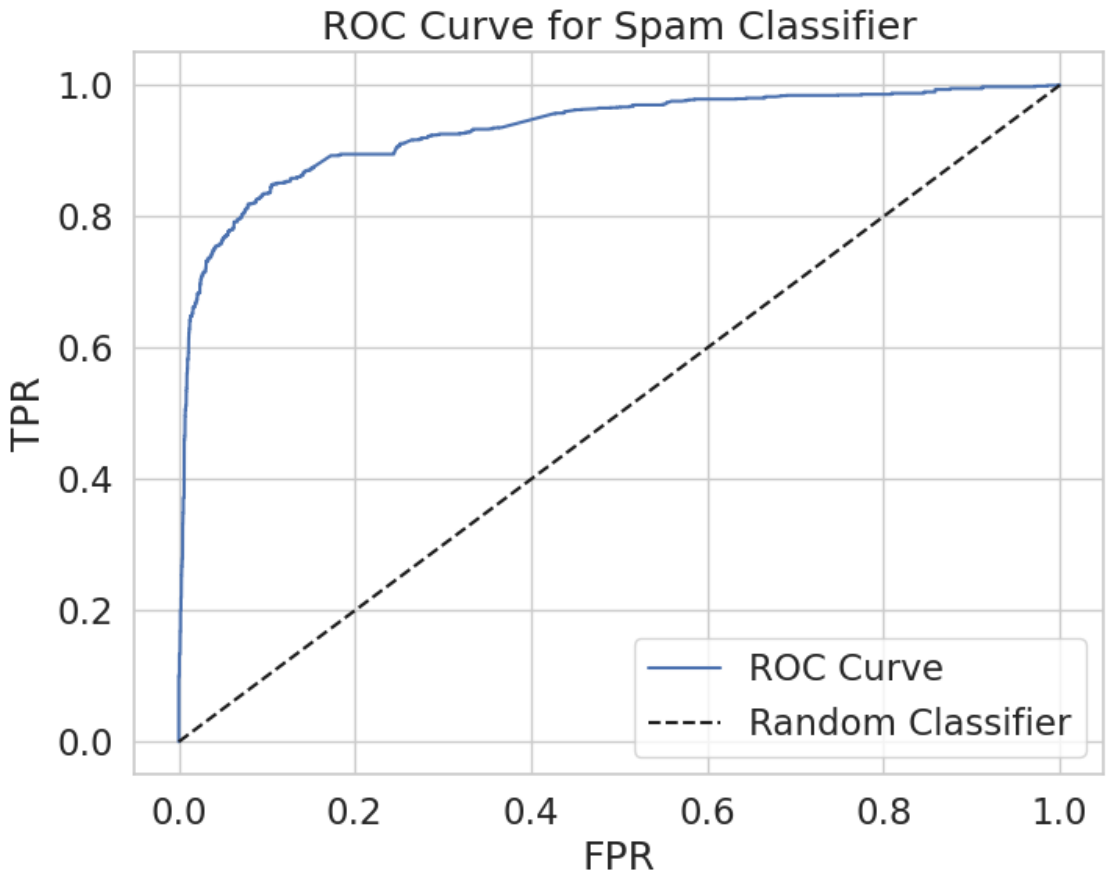
In most cases, we won't be able to get 0 false positives and 0 false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover that they have cancer until 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 Pensieve) 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 [24]: y_probs = my_model.predict_proba(X_train)[:, 1]

         fpr, tpr, thresholds = roc_curve(y_train, y_probs)

         plt.figure(figsize=(8, 6))
         plt.plot(fpr, tpr, label='ROC Curve')
         plt.plot([0, 1], [0, 1], 'k--', label='Random Classifier')
         plt.xlabel('FPR')
         plt.ylabel('TPR')
         plt.title('ROC Curve for Spam Classifier')
         plt.legend()
         plt.grid(True)
         plt.show()
```


ROC Curve for Spam Classifier
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0
ROC Curve
Random Classifier

2.0.1 Question 6a

Pick **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.

When choosing example 3, I would classify this email as spam, because it heavily relies on high-priority triggers from my model's word feature list, such as "save," "offer," "link," and "click." However, others might disagree with this classification as it could be viewed as a legitimate business message about their new update and state that keyword frequency alone cannot always distinguish between spam and ham emails.

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?

The ambiguity lets us realize that the model's predictions don't always result in ground truth, as evaluation metrics like accuracy and precision are simply measured by evaluating the model's ability to identify a valid pattern that is present in some emails. This lets us understand that a misclassified email does not necessarily mean a mathematical failure, but rather a valid interpretation of cases where the model detected a pattern that simply contradicted the human label/ground truth.

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 ID 27. The email changed how email was classified, probably because it was acting as a strong indicator for the spam class. By removing this feature, the email's total weighted log-odds score decreased significantly, causing the predicted probability to fall below the decision threshold. As the predicted probability falls below the threshold, it is now classified as ham email instead of a spam email.

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, because a single feature would not act as a strong indicator for classification. Removing one feature out of 1000 features would result in a tiny change to the total weighted log-odds score and wouldn't be enough to cross the decision boundary.

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.

It would be less interpretable since you won't be able to precisely identify which feature contributes to certain decisions. The weight of the decision will be split across many features.

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.

For the category of "Misinformation," contents like misinformation about health during public health emergencies and misinformation about the dates, locations, times, and methods for voting, voter registration, or census participation would all fall under Facebook's Community Standards of what isn't allowed on Facebook.

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).

Misclassifying a post in the context of social media could affect both the safety of its users and their right to free expression. For example, a false positive would be when a user posts information about the location where voting would be taken, but it was misclassified as misinformation and removed. This would not only restrict the freedom of expression but also prevent others from getting the necessary information on the voting. Moreover, a false negative would mean that if someone posted false information about vaccines being harmful, and would not be removed. This would result in others believing such information and avoid being vaccinated in critical situations.

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 in moderating content as it enables people to easily understand the root cause of certain incorrect classifications and unintended biases. It can be used to debug false positives and understand the precise reasoning behind specific flags, and highlight exactly which part of a post triggered the violation, eventually allowing them to be fixed and have more accurate classifications in the future.