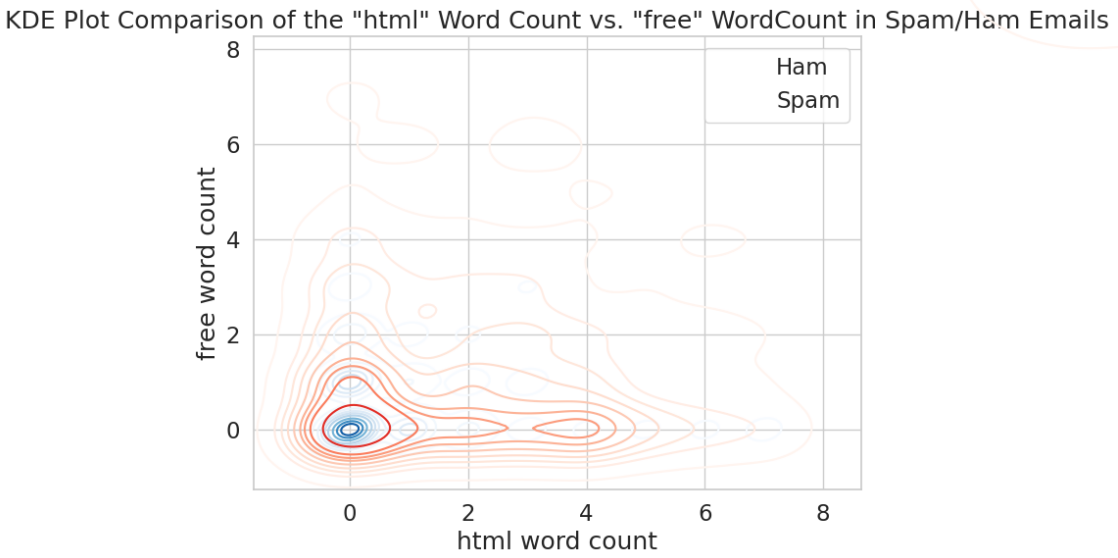

0.1 Question 1a

Generate your visualization in the cell below.

In [82]:
```
email_list = list(original_training_data['email'])
html_word_count = [email.count('html') for email in email_list]
free_word_count = [email.count('free') for email in email_list]

original_training_data['html word count'] = html_word_count
original_training_data['free word count'] = free_word_count

ham = original_training_data['spam']==0
spam = original_training_data['spam']==1

remove_free = original_training_data[original_training_data['free word count'] < 8]
remove_html = remove_free[remove_free['html word count'] < 8]

ham_html = remove_html['html word count'].loc[ham]
ham_free = remove_html['free word count'].loc[ham]

spam_html = remove_html['html word count'].loc[spam]
spam_free = remove_html['free word count'].loc[spam]

plt.figure(figsize=(8,6))
sns.kdeplot(x=ham_html, y=ham_free, cmap='Blues', label='Ham')
sns.kdeplot(x=spam_html, y=spam_free, cmap='Reds', label='Spam')

plt.title('KDE Plot Comparison of the "html" Word Count vs. "free" WordCount in Spam/Ham Emails
plt.legend()
plt.show()
```


KDE Plot Comparison of the "html" Word Count vs. "free" WordCount in Spam/Ham Emails
Ham
Spam
free word count
html word count
0
2
4
6
8

0.2 Question 1b

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

The visualization is a KDE plot comparing how often the words “html” and “free” appear in spam (red) and ham (blue) emails. Spam emails use these words more frequently and across a wider range, while ham emails are concentrated near the origin (0,0). The plot shows that “html” and “free” capture distinct information, making them valuable features for classification, as emails with low counts for these words are more likely to be ham.

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 compared spam and ham emails by plotting features and examining their proportions and distributions. This helped me identify patterns, such as words like "free" and "html" appearing frequently in spam emails. I also focused on structural features, such as "Fw:" and "Re:" in the subject line, which helped capture differences in how spam and ham emails are formatted.

Word-based features like "free" and "offer" worked well because they were clearly more common in spam emails. Structural features like "html" tags also improved the model, as spam emails tend to use more HTML formatting. On the other hand, punctuation-based features like "!" didn't contribute as much as expected, likely because both spam and ham emails use punctuation I guess. However, i would have assumed spam emails to use them more to create a sense of urgency.

I was surprised by how effective structural features like "Fw:" and "Re:" in the subject line were at identifying spam. I didn't expect the formatting to play such a key role, this email syntax was more predictive than I initially thought. It was also interesting to see that small differences in word frequencies for certain features could significantly improve the model's performance.

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.

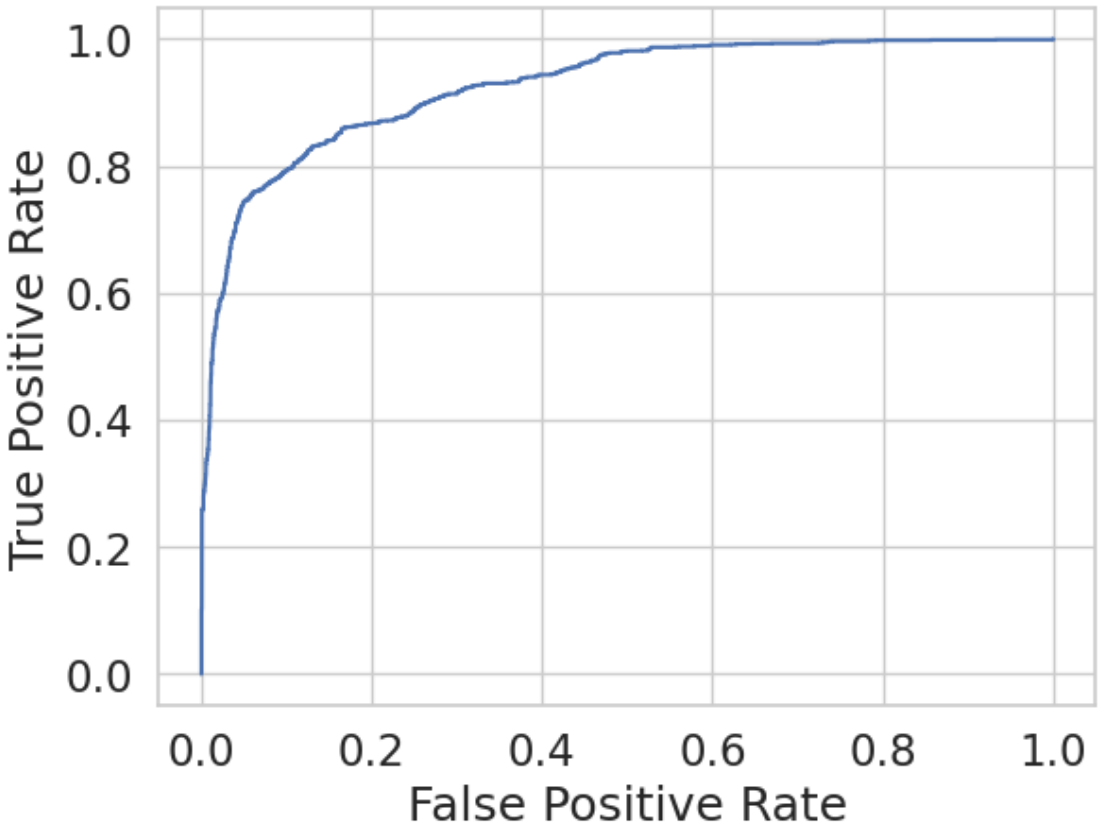
Recall that logistic regression calculates the probability that an example belongs to a particular class. To classify an example, we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, **we can adjust that cutoff threshold**. We can say that an email is spam 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 [96]: from sklearn.metrics import roc_curve
         fpr, tpr, threshold = roc_curve(y_train, model.predict_proba(X_train)[:, 1])
         plt.plot(fpr, tpr)
         plt.xlabel("False Positive Rate")
         plt.ylabel("True Positive Rate")
```

```
Out[96]: Text(0, 0.5, 'True Positive Rate')
```


True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
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 because it is a personal message with no promotional content or call-to-action language typically found in spam. However, the training data labels it as spam. Someone might disagree with my classification because the email contains forwarded content with technical formatting (e.g., mailing list details, links, email headers, punctuations), which could 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 labeled data affects both model predictions and evaluation by introducing uncertainty into the training process and making metrics like accuracy less reliable. When labels are subjective or inconsistent, the model may align with ambiguous patterns rather than true spam or ham characteristics. This makes it harder to fully trust evaluation metrics and highlights the need to account for uncertainty in real-world applications.

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

I removed the email indexed 715 because it contained the least number of words while still including at least one of our features. Although the feature present in the email wasn't the one I chose to remove, its short length and low confidence (55%) that it was spam influenced my decision. I removed the 'bank' feature because it was one of the strongly associated words found in spam emails and changed how email 715 was classified. Without this feature, the model had less confidence in classifying the email as spam, and it flipped to ham, showing how much it relied on 'bank' to make its 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?

In a larger model with 1000 features, it would likely be much harder to find a single feature that changes an email's classification. With so many features, the model's predictions are based on a combination of many subtle signals rather than reliance on a few strongly associated features. Removing one feature would likely have less impact on the overall classification because the remaining features could compensate for the missing information. This increased complexity makes the model more robust but less interpretable at the individual feature level.

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.

The new model with 1000 features would be less interpretable than simple_model. With so many features, it becomes much harder to pinpoint which specific features contribute to an email's classification. In contrast, simple_model is interpretable because it uses a small number of features, making it easier to trace the reasoning behind its predictions. While the new model might be more accurate, its complexity reduces our ability to understand its decision-making process.

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 hate speech, content that attacks individuals or groups based on their race, ethnicity, gender identity, sexual orientation, religion, or disability would fall under this category. According to Facebook's Community Standards, hate speech includes dehumanizing statements, calls for exclusion, or slurs directed at individuals or groups in these categories. I would also add that anything that incites violence against a particular group or portrays a specific community as less than human would fall under this category.

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 hate speech has serious consequences. A false positive (flagging non-hate speech) can suppress free expression and harm trust, especially for marginalized groups discussing their experiences. A false negative (failing to flag hate speech) allows harmful content to remain, potentially leading to harassment or real-world violence.

2.0.5 Question 7e

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

An interpretable model is important because it helps explain why a post is flagged, ensuring decisions align with moderation guidelines. It allows data scientists to spot and fix biases, improve accuracy, and provide transparency to users, which builds trust and accountability.