
0.1 Question 1a

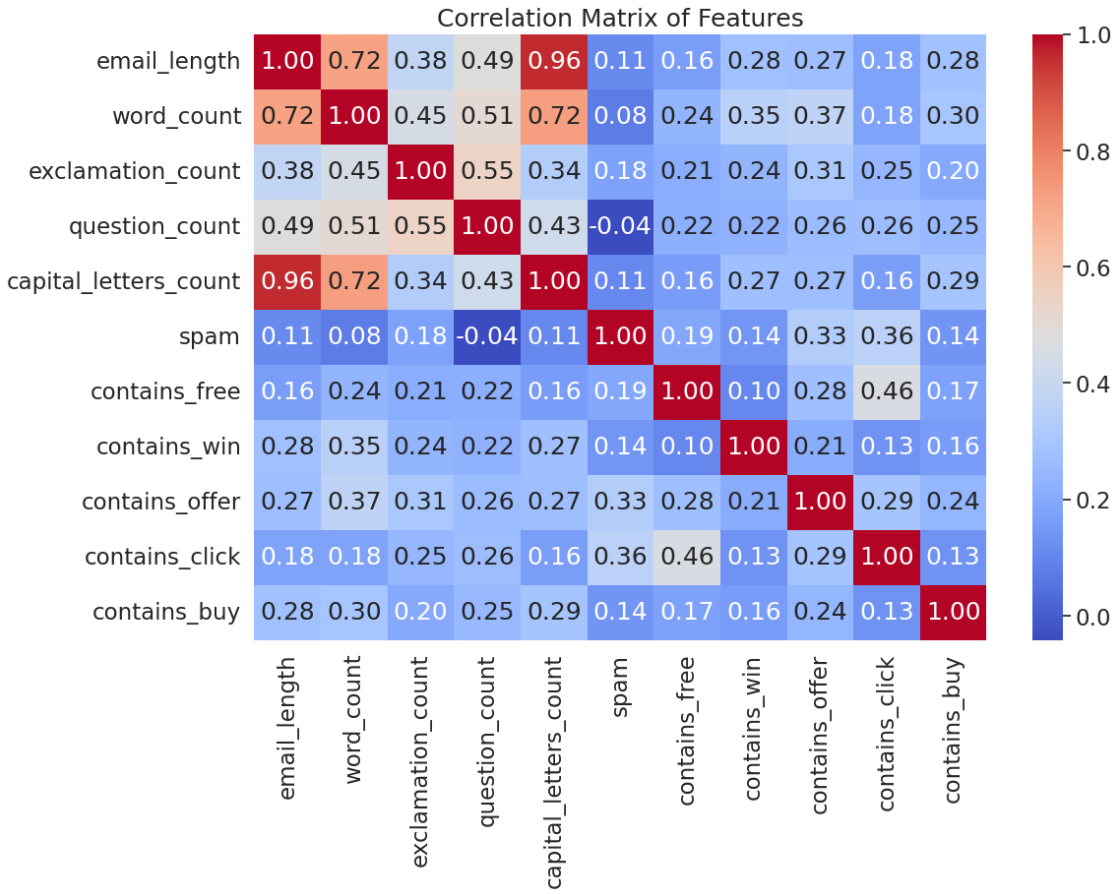
Generate your visualization in the cell below.

In [11]:
```
# I am going to create features that help us pull more meaningful data from the text.
original_training_data['email_length'] =  original_training_data['email'].str.len()
original_training_data['word_count'] = original_training_data['email'].apply(lambda x: len(x.sp
original_training_data['exclamation_count'] = original_training_data['email'].apply(lambda x: x
original_training_data['question_count'] = original_training_data['email'].apply(lambda x: x.co
original_training_data['capital_letters_count'] = original_training_data['email'].apply(lambda

# Pull out most common SPAM words as from 1B.
spam_keywords = ['free', 'win', 'offer', 'click', 'buy']
for word in spam_keywords:
    original_training_data[f'contains_{word}'] = original_training_data['email'].apply(lambda x

correlation_matrix = original_training_data[['email_length', 'word_count', 'exclamation_count'
                                  'question_count', 'capital_letters_count', 'spam'] +
                                 [f'contains_{word}' for word in spam_keywords]].corr()

plt.figure(figsize=(12, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Features')
plt.show()
```


Correlation Matrix of Features
email_length 1.00 0.72 0.38 0.49 0.96 0.11 0.16 0.28 0.27 0.18 0.28
word_count 0.72 1.00 0.45 0.51 0.72 0.08 0.24 0.35 0.37 0.18 0.30
exclamation_count 0.38 0.45 1.00 0.55 0.34 0.18 0.21 0.24 0.31 0.25 0.20
question_count 0.49 0.51 0.55 1.00 0.43 -0.04 0.22 0.22 0.26 0.26 0.25
capital_letters_count 0.96 0.72 0.34 0.43 1.00 0.11 0.16 0.27 0.27 0.16 0.29
spam 0.11 0.08 0.18 -0.04 0.11 1.00 0.19 0.14 0.33 0.36 0.14
contains_free 0.16 0.24 0.21 0.22 0.16 0.19 1.00 0.10 0.28 0.46 0.17
contains_win 0.28 0.35 0.24 0.22 0.27 0.14 0.10 1.00 0.21 0.13 0.16
contains_offer 0.27 0.37 0.31 0.26 0.27 0.33 0.28 0.21 1.00 0.29 0.24
contains_click 0.18 0.18 0.25 0.26 0.16 0.36 0.46 0.13 0.29 1.00 0.13
contains_buy 0.28 0.30 0.20 0.25 0.29 0.14 0.17 0.16 0.24 0.13 1.00
email_length
word_count
exclamation_count
question_count
capital_letters_count
spam
contains_free
contains_win
contains_offer
contains_click
contains_buy
1.0
0.8
0.6
0.4
0.2
0.0

0.2 Question 1b

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

The heat map displays the relationship between the various attributes of the email dataset, including the number of capital letters, exclamation points, and email length, as well as particular keywords that I chose after researching the most frequently used terms in spam emails. The heat map shows that some keywords, like "free," "offer," and "click," appear to be more associated with spam emails than other elements. Overall, though, it doesn't appear that the attributes I've developed have a strong link with spam emails—correlation values should be closer to 1.0. Therefore, as I work through the project's subsequent revisions, I should improve my model's accuracy by adding more features and better capturing what constitutes a spam email.

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. After determining the connection between the different email attributes and my goal variable (spam or ham), I found features that were more strongly associated with and biased toward spam emails.
2. It really performed rather well when we first tried adding features based just on the existence of common terms, like we did with 2A, as well as the email identifiers (email length, capitalization, etc.). It was insufficient, nevertheless, to meet the accuracy threshold of 80%. I therefore made the decision to follow the guidance the project continued giving me and incorporate a wider variety of characteristics, such the email structure (the html elements we covered in project 2A). This enhanced the model's performance, but I was still just below the 85% mark. This, in my opinion, was caused by the excessive number of identical features I introduced, which resulted in a significant amount of multicollinearity. I therefore made the decision to select more significant features in order to address this problem. I added a larger variety of trigger words and ended upincreasing my models accuracy from 79% to a whopping 87%. Given i struggled with themodel developement in project 1A this made me feel extremely happy.
3. What surprised me the most about the criteria used to evaluate a feature was how some trigger words affected the ability to identify spam emails. Actually, expanding my database of searchable words—rather than adding HTML tag captures—was the biggest improvement I saw in model accuracy. This surprised me because I thought that producing incredibly complex features was the key to increasing accuracy. Instead, it was far simpler than I had initially thought.

2 Question 5: ROC Curve

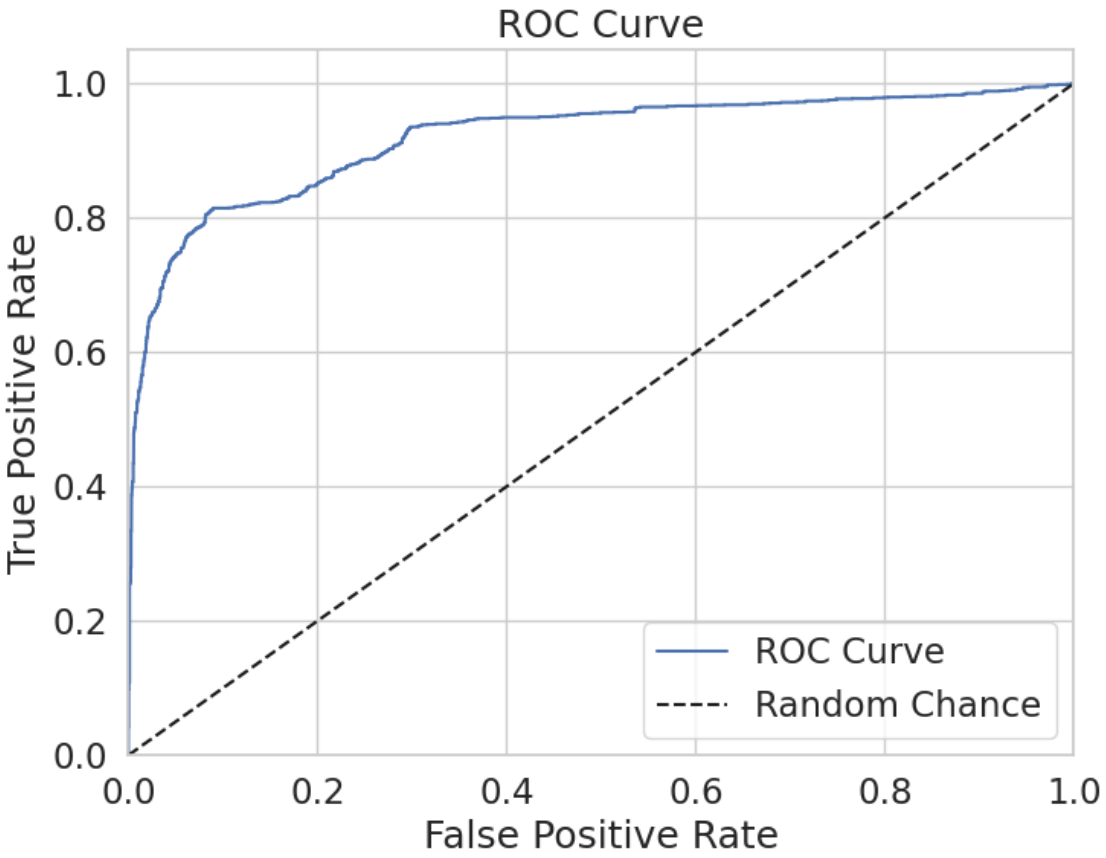
In most cases, we won't be able to get 0 false positives and 0 false negatives, so we have to compromise. For example, in the case of cancer screenings, false negatives are comparatively worse than false positives — a false negative means that a patient might not discover that they have cancer until 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 [18]: probabilities = model.predict_proba(X_train)[:, 1]
         fpr, tpr, _ = roc_curve(Y_train, probabilities)
         # Plot the ROC curve
         plt.figure(figsize=(8, 6))
         plt.plot(fpr, tpr, label='ROC Curve', color='b')
         plt.plot([0, 1], [0, 1], 'k--', label='Random Chance')
         plt.xlim([0.0, 1.0])
         plt.ylim([0.0, 1.05])
         plt.xlabel('False Positive Rate')
         plt.ylabel('True Positive Rate')
         plt.title('ROC Curve')
         plt.legend(loc='lower right')
         plt.show()
```


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

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.

MR. Mabo Loko's initial email in example 1 talks about a "urgent businessproposal" that involves an inheritance scam. This is in line with the training data's classification as spam. Although it seems rare, someone could disagree with me. The email is obviously a phishing scam, even though it might be a legitimate business proposition and they might be attempting to take advantage of you to profit financially. The email asks for sensitive personal information from us as the user and makes extravagant financial promises. The algorithm should appropriately recognize the spam email as such since, aside from its official tone, it is much more likely to be a fraud or phishing attempt.

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?

Our comprehension of model predictions is impacted by the ambiguity of laveled data in data science since it introduces bias. The bias is that we frequently train our models on data that we perceive to represent the ground truth. Occasionally, there may be differences between the model's predictions and actual human judgment because we train and assess our models using said data. To put it another way, even while the models are showing high accuracy, the way the data is interpreted may cause the models to either overestimate or underestimate the subjective accuracy in a particular scenario. Therefore, as data scientists, our first task is to ensure that our labeled data is as free of potential biases as possible. This will help to mitigate the impact of these biases on our final product and increase the TRUE efficacy of the model.

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 word "bank" probably plays a big part in phishing scams, which is why I think the removal of the word altered the email's classification. It is even more evident from question 6 that banks, financial organizations, and other relevant sectors are heavily involved in phishing scams. When creating models that use words as classification characteristics, this might be problematic because many users will get emails containing urgent financial information; if those items are mistakenly categorized as spam, the user could suffer greatly. Because there were probably less comments in the email that suggested it was spam, we could observe that the models' confidence in the email being ham decreased considerably after we eliminated the bank.

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, the reason why is because as the complexity of features increases the weight that a specific feature carries is diluted. The features importance in the overall model thus would have less of a down stream affect than a model that contains 50-100 features. To further reiterate, as the number of features n increases to a larger number » the overall prediction contribution of a specific feature is smaller in proportion to the entire model. In contrast, as the number of features n decreases to a smaller number « the overall prediction contribtuion of a specific feature increases and thus the removal of a feature can be far more significant that a more complex model running 100s if not 1000s of features

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.

There may be advantages and disadvantages to a model with a lot of features. It would undoubtedly be harder to understand than the simple_model, even though it might be more correct in some circumstances. This is because it becomes harder to interpret the overall specific contribution of a particular feature as models get more complicated. Therefore, it would be simpler for us to see how the removal of a particular model would affect accuracy or confidence in this case of comprehending models' contribution to email classification than if a model had 1000 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.

While it is rather simpler to categorize hate speech, violence, and incitement. Since the development of such a model can be rather interesting and, in my opinion, more complex, I thought misinformation would be a bit more interesting to choose when discussing a moderation model. This is because some posts may discuss ideas that are not directly stated by the public or that are common knowledge, so how could a post be accurately categorized as a misinformation post? False or purposefully misleading information, as well as the use of misleading information to intentionally or inadvertently deceive others, are all considered misinformation according to Facebook Community Standards. These include false information about events, claims about public health (like new viruses or conspiracy weapons to incite public hysteria), and manipulated or artificially created visual content to mislead people (like deepfakes of well-known public figures to promote cryptocurrencies or another type of business model).

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).

A false positive happens when a post is incorrectly flagged as misinformation even though it's actually accurate. In the context of misinformation, this kind of mistake can infringe on free speech and limit the spread of truthful content. That not only harms the public's access to accurate information, but also damages users' trust in the platform. If people feel their honest posts are being unfairly labeled or taken down, they may begin to question the platform's fairness and credibility.

A false negative is when a post containing misinformation is wrongly classified as true and left up. This can be dangerous, especially when the misinformation spreads quickly and influences public opinion. For example, after the attempted assassination of Donald Trump, some posts falsely claimed it was a planned attack by the Democratic Party. If those kinds of posts aren't caught, they could lead to serious consequences like public outrage, radicalization, or even violence. That's why it's critical to reduce both false positives and false negatives — to protect free speech while also keeping the platform safe and trustworthy.

2.0.5 Question 7e

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

As a data scientist, having an interpretable model is really important when moderating content online because it helps us understand why a certain post was flagged. If a user's content gets taken down or labeled, we need to be able to explain the reason — especially when the content is borderline or controversial. Interpretable models allow us to trace decisions back to specific features, like the presence of certain words or phrases. This transparency helps build trust with users, makes it easier to identify and fix errors, and allows for better accountability in content moderation decisions. It also helps ensure that our models aren't making biased or unfair choices based on hidden patterns we don't fully understand.