0.0.1 Question 1: Feature/Model Selection Process

In this following cell, describe the process of improving your model. You should use at least 2-3 sentences each to address the follow 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 tried to find features based on the recommendations given. First I attempted the usage of more/better words.Checking through our emails, I first considered words with the highest count. This however was not helpful because many of the common english words are present in both sets, so we did have to conduct a different count of words. Then I merged our two dataframes, one with spam emails where 0 indicates it is not a spam email and 1 which indicates they are spam emails. I created a new column called difference which computes the difference between words used in spam and words used in ham. I noticed that spam words tend to have more undefined pattern and generated using symbols such as "!" and "<" whereas ham emails do not contain much of that when looking at the positive difference. My second experiement involving looking at emails and determing my selection of words I felt would not be in ham based on writing respectful emails. Words such as sex,money,!,etc raised more spam emails when graphing the proportions. The presence of certain words made a difference when considering the psychology of writing a respectful decent email compared to spam. After that, we just iterate through words, I took clusters of 100 using iloc, so for example the first 100 entries would be :100, 100:200, and so on. Doing this trial and error process we were able to deduce the difference in words between spam and ham and thus find better features for our model.

2.) Some of the features selected proved to be very useful such as the proportion of provactive terms in spam vs ham emails. The existence of symbols also helps to separate our spam from ham emails. However finding the count of words is not very useful because in the english language there are so many commonly used words that is present in both spam and ham. Using more words impacted the test accuracy, but we still have to consider overfitting our data, however adding more terms means better test accuracy. What didnt work for me was just using .head(), and looking at the text of spam and ham emails, through eyeballing it we gradually receive a better estimate, though it is not enough to warrant great accuracy, and often words are found in both. Finding the absolute difference is also a possiblity because if we only consider positive numbers we limit difference between spam and ham, there might be cases where ham have more than spam words.

3.) A few surprising aspects arose from my two experiments. First, spam emails have more symbols compared to ham though the difference is not that significant . The presence of more greedily and provactive used terms arise from spam emails compared to ham. How one determines them is very subjective and it would be beneficial to form an algorithm that captures this group very well.I found surprising however many funny terms positively impacted and increased our test accuracy. Symbols really impacted the accuracy of my model, adding a few resulted in greater accuracy when coupled with some provactive and insensitive terms.

Question 2a Generate your visualization in the cell below.

In [51]:
```
#@source https://stackoverflow.com/questions/26778899/how-to-convert-matrix-to-pandas-data-fra

words_sample = ['sex','penis','money','rich','promotion','cash','!!','%','align',
        'font','<td','millionaire','god','fast','guarantee',
        'top','bank','refund','trust','scam','affordable',
        '=','your']

new_train2 = train
for i in words_sample:
    new_train2[i] = words_in_texts([i],train['email'])

plot_correlation_words = new_train2[words_sample]
plot_correlation_words
```

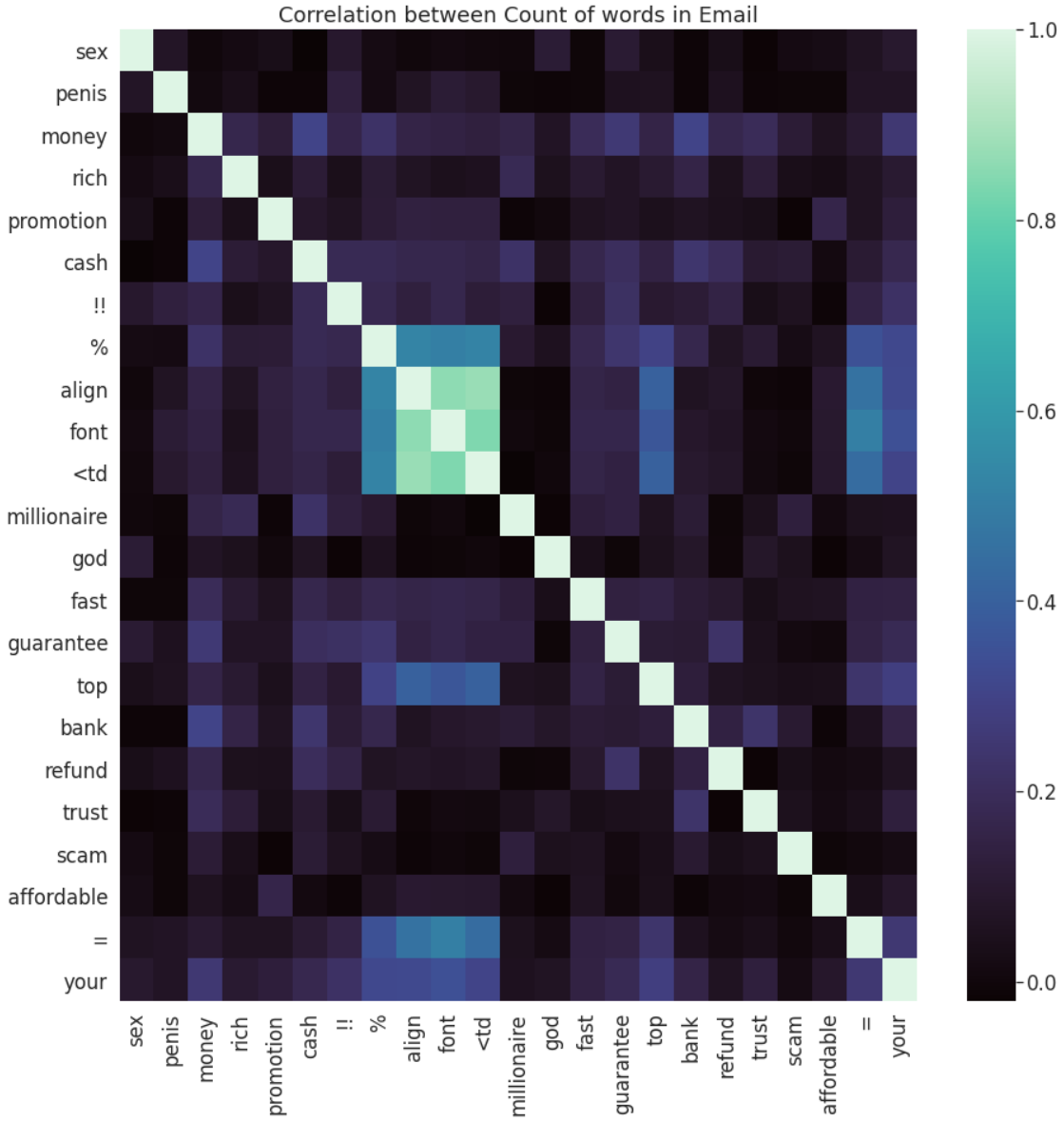
Out[51]:

	sex	penis	money	rich	promotion	cash	!!	%	align	font	…	fast	\
0	0	0	0	0	0	0	0	0	0	0	…	0	
1	0	0	0	0	0	0	0	1	0	0	…	0	
2	0	0	0	0	0	0	0	0	0	0	…	0	
3	0	0	0	0	0	0	0	0	0	0	…	0	
4	0	0	0	1	0	0	0	1	1	1	…	1	
…	…	…	…	…	…	…	.. ..	…	…	…	…		
7508	0	0	0	0	0	0	0	0	0	0	…	0	
7509	0	0	1	0	0	0	1	0	0	0	…	0	
7510	1	0	0	0	0	0	0	0	0	0	…	0	
7511	0	0	0	0	0	0	0	0	0	0	…	0	
7512	0	0	0	0	0	0	0	0	0	0	…	0	

	guarantee	top	bank	refund	trust	scam	affordable	=	your
0	0	1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	1	0
2	0	0	0	0	0	0	0	1	0
3	0	1	0	0	0	0	0	1	0
4	0	1	0	0	0	0	0	1	1
…	…	…	…	…	…	…	… ..	…	
7508	0	0	0	0	0	0	0	1	0
7509	0	0	1	0	0	0	0	1	1
7510	0	0	0	0	0	0	0	0	1
7511	0	0	0	0	0	0	0	0	0
7512	0	0	0	0	0	0	0	0	0

[7513 rows x 23 columns]

In [52]:
```
fig, ax = plt.subplots(figsize=(15,15))
sns.heatmap(plot_correlation_words.corr(),cmap="mako");
plt.title("Correlation between Count of words in Email");
```


Correlation between Count of words in Email
sex
penis
money
rich
promotion
cash
!!
%
align
font
<td
millionaire
god
fast
guarantee
top
bank
refund
trust
scam
affordable
=
your
1.0
0.8
0.6
0.4
0.2
0.0

Question 2b Write your commentary in the cell below.

The following as suggested is a heatmap plotting the correlation between the number/count of the word appearing. The following was chosen with a "mako" color because the defeat colors involves purple which is difficult to see the correlation at colors for example between 0.8 and 1 since the colors are too similiar. With the mako, we can better tell the difference between correlation of certain words. The plot itself shows the relationship between the following words. For example, there is a large correlation between "=" chaarcter and "font" suggesting these words are used together. Words with higher correlation can be used to better tell us which words in our following spam model will make our predictions more accurate.

0.0.2 Question 3: 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, whereas a patient can just receive another screening for a false positive.

Recall that logistic regression calculates the probability that an example belongs to a certain class. Then, 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*: 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 ROC curve shows this trade off for each possible cutoff probability. In the cell below, plot a ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Refer to Lecture 20 to see how to plot an ROC curve.

Hint: You'll want to use the `.predict_proba` method for your classifier instead of `.predict` so you get probabilities instead of binary predictions.

```
In [53]: test_words_for = ['!','money','rich','cash','!!!','%','align',
                           'height','url','body','<','drug','penis','boner',
                           'kiss','love','<html>','boob','bank','swag','>','=',
                           'font','<td','millionaire','god','fast','guarantee',
                           'top','CEO','bank','refund','trust','scam','affordable','=',
                           'your']
         X_train2 = words_in_texts(test_words_for, train['email'])
         Y_train2 = np.array(train['spam'])
         X_test_creation = words_in_texts(test_words_for, test['email'])
         model = LogisticRegression(solver = 'lbfgs')
         model.fit(X_train2,Y_train2)


Out[53]: LogisticRegression()


In [67]: from sklearn.metrics import roc_curve

         y = train['spam']
         y_predict = model.predict_proba(X_train2)
         y_predict2 = y_predict[:,1]
         fpr, tpr, threshold = roc_curve(y,y_predict2)

         fig = sns.lineplot(x=fpr,y=tpr)
         fig.set_xlabel('False Positive Rate')
         fig.set_ylabel('True Positive Rate')
         fig.show()
```

```
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_264/928421770.py in <module>
      8 fig = sns.lineplot(x=fpr,y=tpr)
      9 fig.set_xlabel('False Positive Rate')
---> 10 fig.set_ylabel(title='True Positive Rate')
     11 fig.show()

TypeError: set_ylabel() missing 1 required positional argument: 'ylabel'
```