0.1 Question 1

In the following cell, 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?

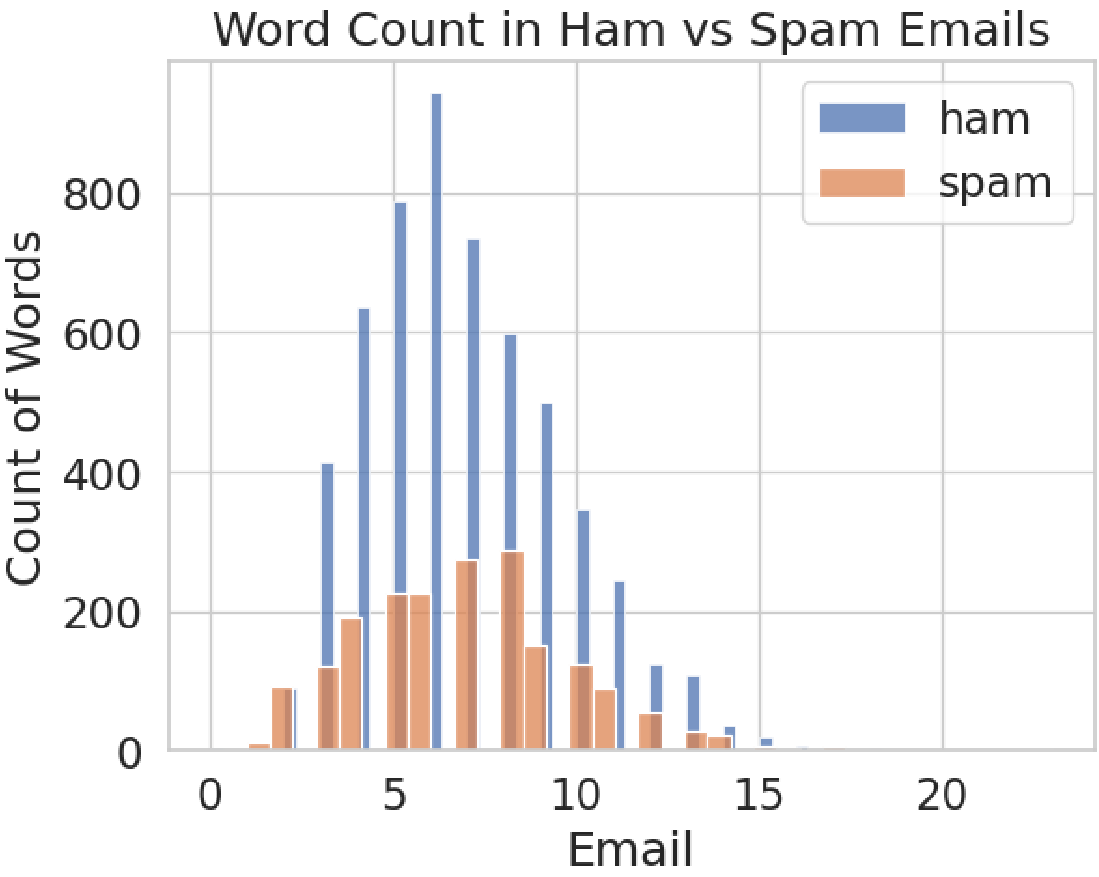
1. I found better features in my model by going through the 'spam' list in the emails. By going one by one through the spam emails one by one and seeing common overlaping words. While this was time consuming since I had to go through them one by one in jupyternotebooks, it seemed like the bestway.
2. So far, I have not done anything to the original data, just selected a nice set of words and common things in spam emails has gotten me to over 85% in the training set. Revision: After seeing that this did not get me a high enough score on the test set I have decided to compare the word count of the Ham vs Spam emails and revsie my word list.
3. I found that some words that I thought would be good for my training model such as 'big' or 'now' actually ended up hurting my model even though I saw it in a lot of spam emails. This is just me not looking at the ham emails for a better sense of the data.

0.2 Question 2a

Generate your visualization in the cell below.

```
In [59]: sns.histplot(train[train['spam']==0]['subject_filter'], label = 'ham')
         sns.histplot(train[train['spam']==1]['subject_filter'], label = 'spam')
         plt.legend()
         plt.title('Word Count in Ham vs Spam Emails')
         plt.xlabel('Email')
         plt.ylabel('Count of Words')
```

```
Out[59]: Text(0, 0.5, 'Count of Words')
```



0.3 Question 2b

Write your commentary in the cell below.

Here I saw that ham emails in general tend to be more verbose and seem to contain more substance. Spam emails seem to have far fewer words, this is most likely due to the attempt to try and catch the readers attention as fast as possible.

0.4 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. In contrast, a patient can receive another screening for a false positive.

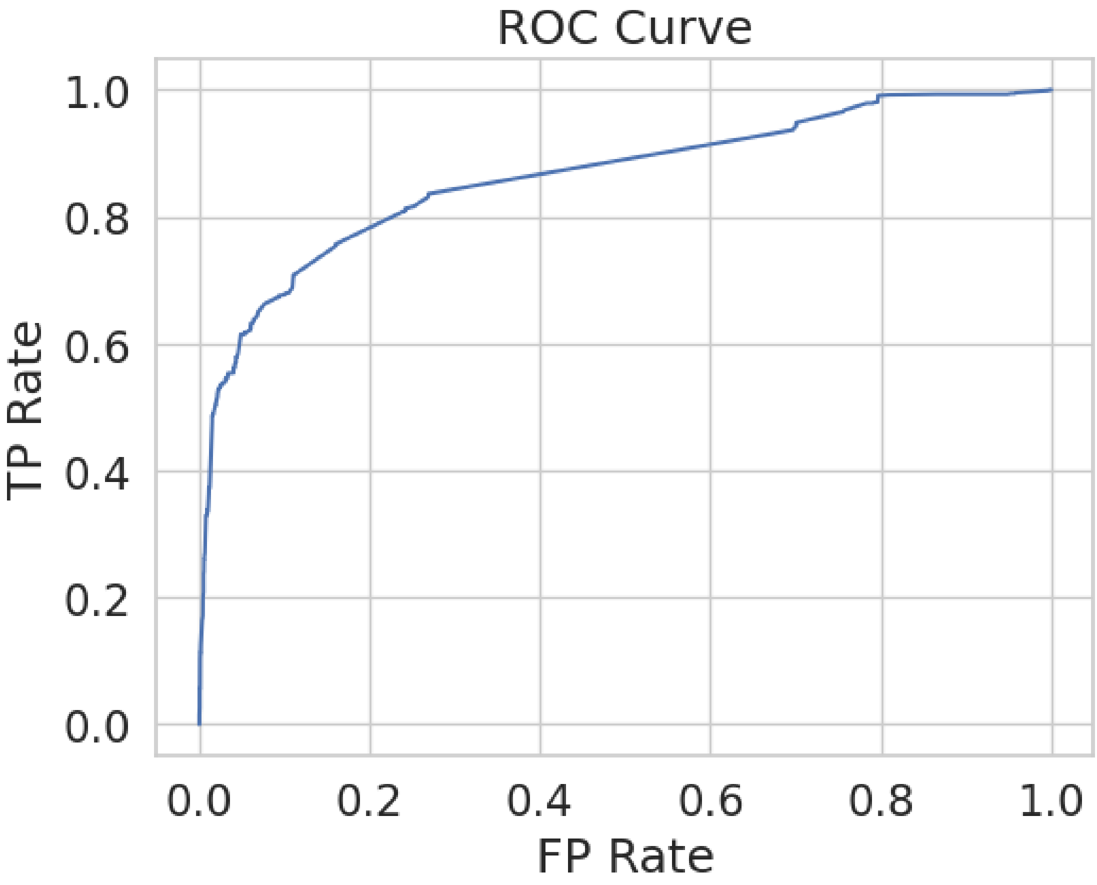
Recall that logistic regression calculates the probability that an example belongs to a particular class. To classify an example, we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, **we can adjust that cutoff threshold**: We can say that an email is spam 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. Refer to Lecture 23 to see how to plot an ROC curve.

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

```
In [60]: fpr, tpr, _ = roc_curve(Y_train, model.predict_proba(X_train)[:, 1])
         plt.plot(fpr, tpr)
         plt.title('ROC Curve')
         plt.ylabel('TP Rate')
         plt.xlabel('FP Rate')
```

```
Out[60]: Text(0.5, 0, 'FP Rate')
```


ROC Curve
TP Rate
FP 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