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. Initially I followed all the suggestions like finding the percent of capital letter in subject etc. But I only got around 50% accuracy for the validation set. I then decided to apply all these filters for both the subject and email. And it also took a significant amount of time to find the most appeared word in spam and ham emails to decide on the best words to use.

2. I thought that num punc would be very efficient, but I saw some emails that are not spam has very high number of punctuations it turned out that it was written in html thus detected lots of non characters. Did some html extraction and it improved the accuracy. Also, the number of punctuations really does not matter, because the length of emails varies, so I tried proportion of punctuations intead.

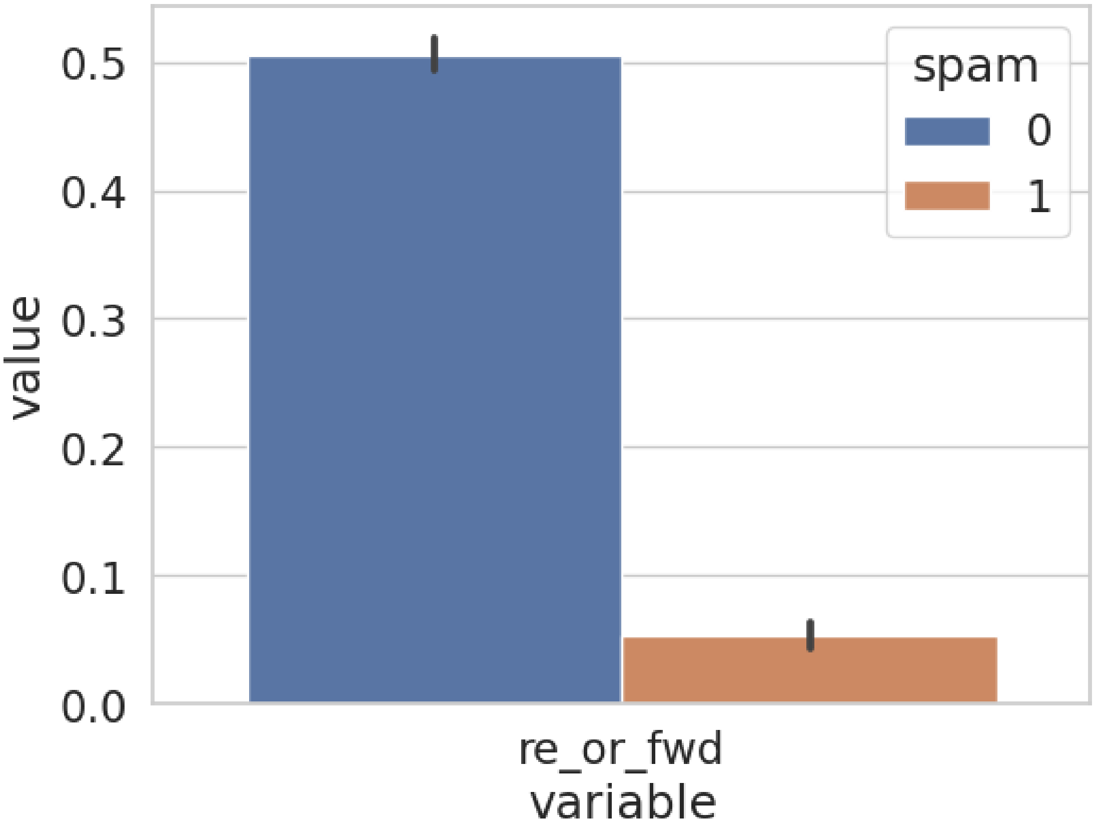
3. Some words are surprisingly very frequently used in spam email like "you", "to", "and". Try including number of words in the email. Does not improve the model.

0.2 Question 2a

Generate your visualization in the cell below.

```
In [72]: melt_re_or_fwd = new_train[['spam', 're_or_fwd']].melt('spam')
         sns.barplot(x = melt_re_or_fwd['variable'], y = melt_re_or_fwd['value'], hue = melt_re_or_fwd[
```

```
Out[72]: <Axes: xlabel='variable', ylabel='value'>
```



0.3 Question 2b

Write your commentary in the cell below.

Using the melt function introduced from B1, I comparethe proportion of spam and ham emails that are either fowarded email or replies. For the first visualization there is a clear difference in height between spam and ham emails, that emails that are either fowarded email or replies are most likely to be ham emails.

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.

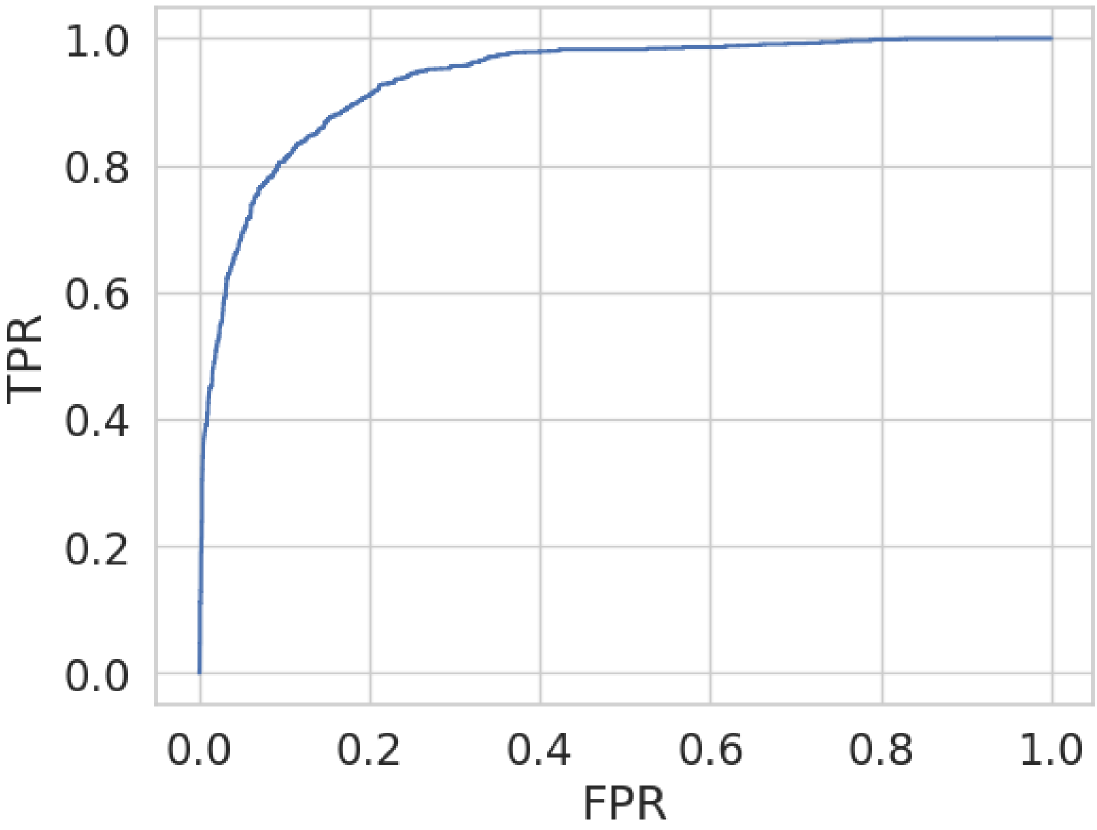
Recall that logistic regression calculates the probability that an example belongs to a particular class. To classify an example, we say that an email is spam if our classifier gives it ≥ 0.5 probability of being spam. However, **we can adjust that cutoff threshold**: We can say that an email is spam 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 [73]: p = model.predict_proba(X_train)[:, 1]
         #display(model.classes_)
         fprs, tprs, T = roc_curve(Y_train, p)

         plt.plot(fprs, tprs)
         plt.xlabel("FPR")
         plt.ylabel("TPR");
```


1.0
0.8
0.6
0.4
0.2
0.0
TPR
0.0
0.2
0.4
0.6
0.8
1.0
FPR