0.0.1 Question 1: Feature/Model Selection Process

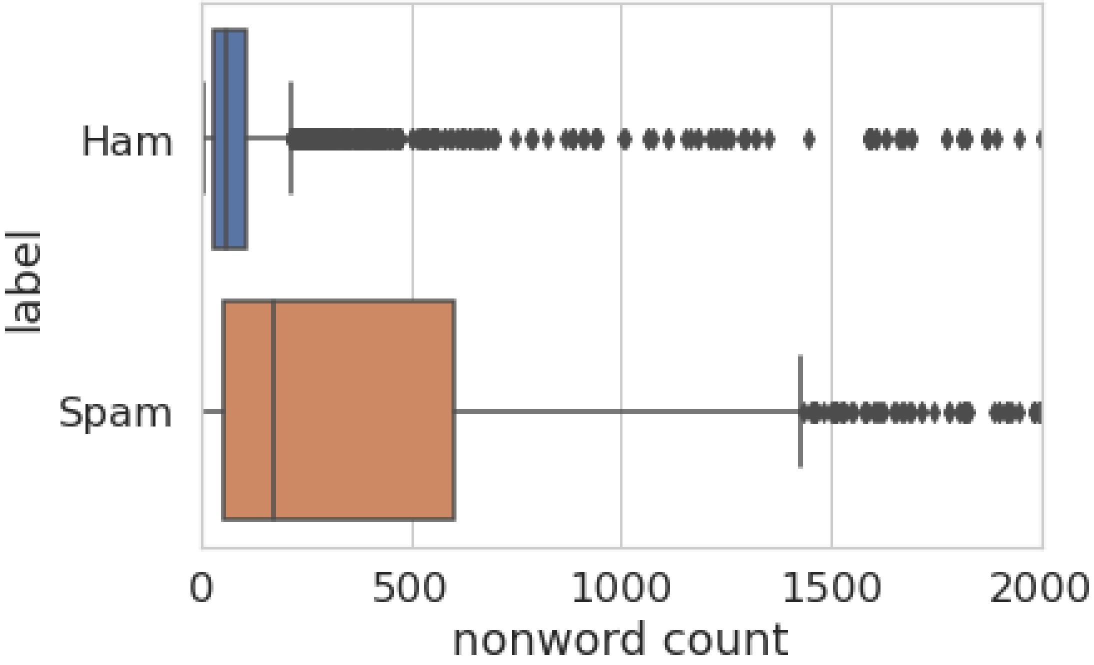
In this following cell, describe the process of improving your model. You should use at least 2-3 sentences each to address the follow questions:

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) To find better features for my model, I looked into different features that would help distinguish spam and ham. To do this, I looked at various features and their relaitonship with spam/ham by looking at their proportions and distributions. If there is a visible and clear difference that distinguishes spam and ham, I will be adding that feature. I tried all the recommended ones by the guide and I looked for the most common words in either spam or ham.
2) Something that worked was that features I used such as number of exclamation marks, had different distribution for the spam and ham. This worked well as a feature. Something that didn't work was that the proportion and distributions for the different features were all similar if we looked at Subject. This made it so that features relating to the Subject were not helpful and should not be used.
3) Something I found interesting was that ham contained few HTML tags compared to spams which has a distribution that was more spread out and had more HTML tags. However, it was interesting to see this to be the opposite for the URL distributions. Spams had little to no URLs while hams had more and was spread out. These two can be used as features.

Question 2a Generate your visualization in the cell below.

```
In [298]: trains = train.copy()
          trains['nonword count'] = train['email'].str.findall('[^\sa-zA-Z0-9]+').apply(len).fillna(0)
          trains['label'] = trains['spam'].replace({0:'Ham', 1:'Spam'})
          plt.xlim(0,2000)
          sns.boxplot(data=trains, x='nonword count', y='label');
```



Question 2b Write your commentary in the cell below.

One way to visually depict whether spam emails tend to have more nonwords than ham emails is to create a boxlplot of the distribution of Spam/Ham and the number of nonword characters in their email body. Above I did just that. As we can see in the two boxplot distributions, the distrubution for Ham tends to be more dense at lower amounts of nonwords, with the median at about 50 words. The distribution for Spam is more spread out with its median being higher than Ham at about 175. Since the distribution shows that spams tend to have more nonwords than hams (sort of), adding nonword count of email body may be beneficial to our model.

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.

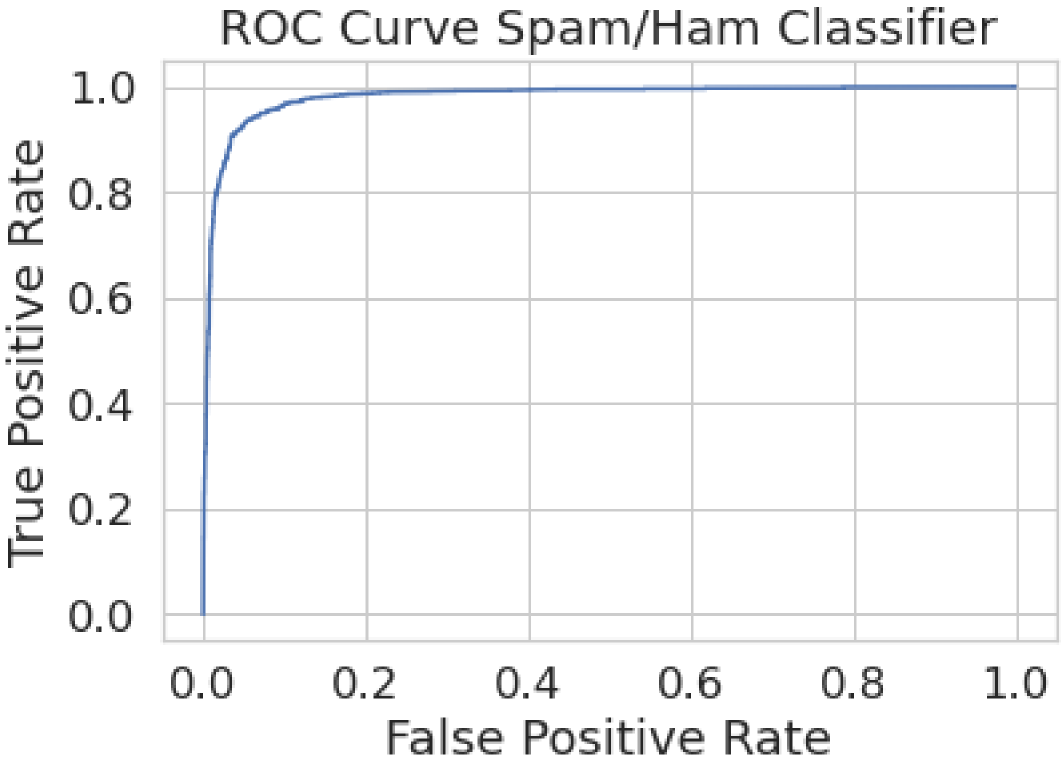
The ROC curve shows this trade off for each possible cutoff probability. In the cell below, plot a ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Refer to Lecture 20 to see how to plot an ROC curve.

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 [300]: from sklearn.metrics import roc_curve

          predictions = model_new.predict_proba(X_train)[:, 1]
          fpr, tpr, thresholds = roc_curve(Y_train, predictions)

          plt.plot(fpr, tpr)
          plt.xlabel('False Positive Rate')
          plt.ylabel('True Positive Rate')
          plt.title('ROC Curve Spam/Ham Classifier');
```


ROC Curve Spam/Ham Classifier
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate