
0.1 Question 1

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?

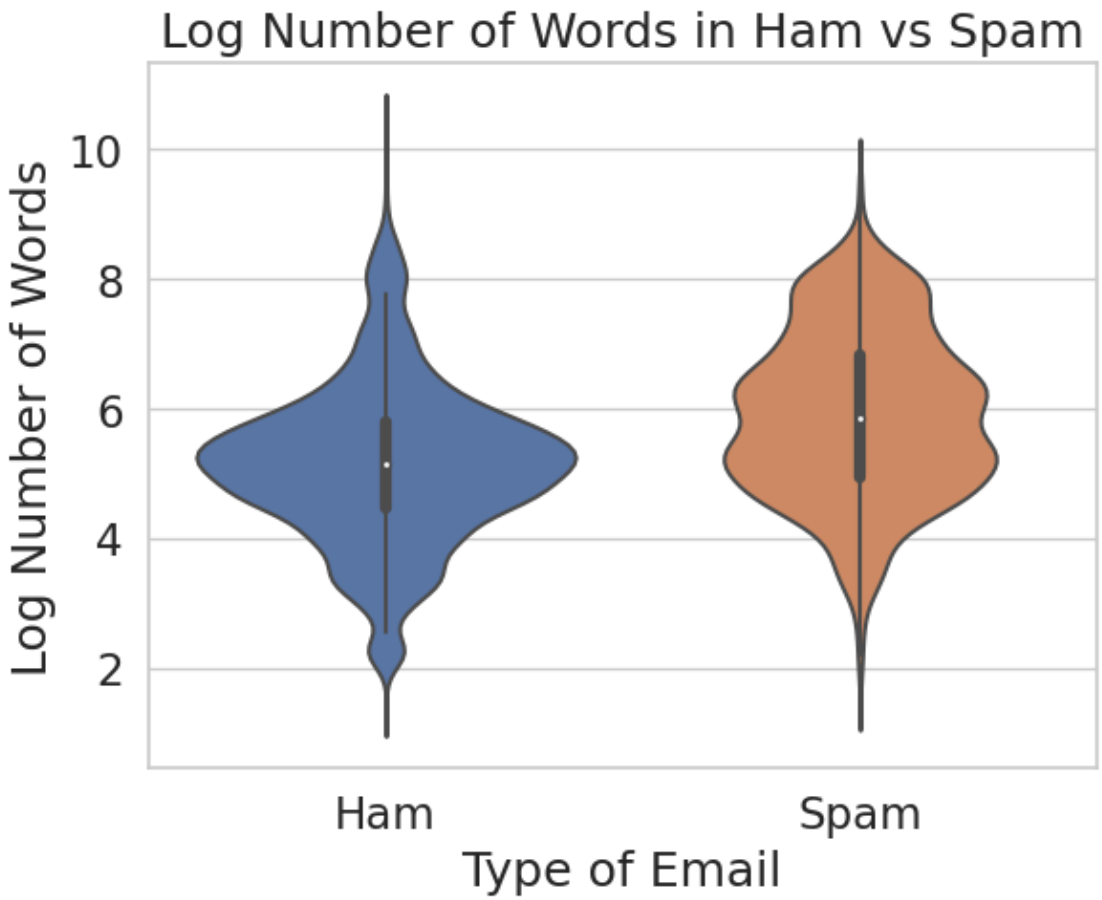
The first thing I did was look up words that google uses to calssify spam. This did not work as well as I thought. After that I went through the examples given and made those features. I would combine some like the number of words in the subject and body and compare this to countung them seperately. I also tried to find which words appear more in spam than in ham. I did this by counting each word in spam and ham and then subtracting the words in ham count from the words in spam count. The top words appreaded more frequently than in ham, so I used those words. I also checked the coefficients of my features, but I found out this was a terrible idea after going to office hours and talking to tutors. I then decided to think about the features more and decided which ones probably mattered the most. Like the number of characters/ words in the subject did not matter that much.

0.2 Question 2a

Generate your visualization in the cell below.

```
In [50]: df = train.copy()
         df['body_word'] = np.log(df['email'].str.split(' ').apply(len))
         df['spam'] = df['spam'].replace(
                                          {0: "Ham",
                                           1: "Spam"}
                                         )
         sns.violinplot(data= df,x = 'spam', y='body_word')
         plt.ylabel('Log Number of Words')
         plt.xlabel("Type of Email")
         plt.title('Log Number of Words in Ham vs Spam')
```

```
Out[50]: Text(0.5, 1.0, 'Log Number of Words in Ham vs Spam')
```



0.3 Question 2b

Write your commentary in the cell below.

I made a plot showing the log of the number of words in the spam and ham. The unlogged number was too big wich made the plot really hard to read. I wanted to see which had more words and it seems like a vast majority of ham have the same amout of words. It has one peak, while I would say spam has 3 peaks. This means that spam emails are not usually around the same number of words.

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, 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 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.

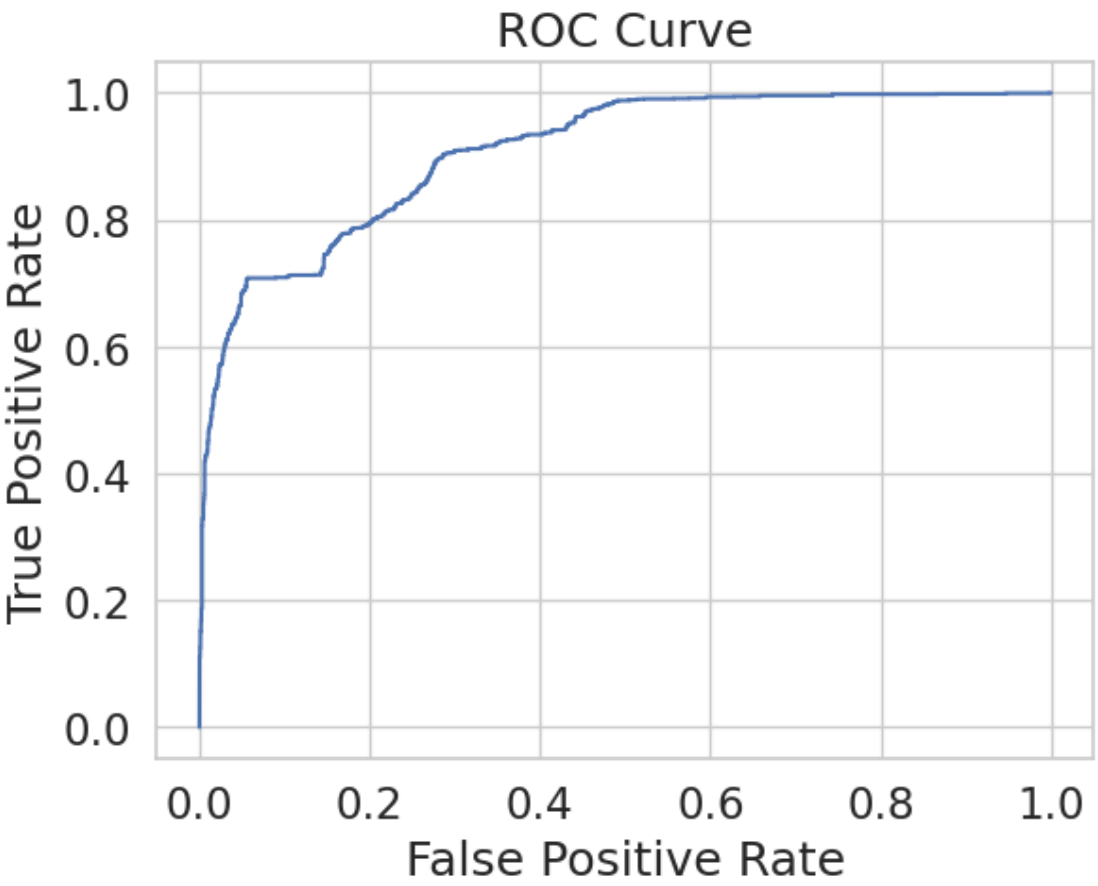
The Receiver Operating Characteristic (ROC) curve shows this trade off for each possible cutoff probability. In the cell below, plot a ROC curve for your final classifier (the one you use to make predictions for Gradescope) on the training data. Refer to Lecture 24 to see how to plot an ROC curve.

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

```
In [51]: from sklearn.metrics import roc_curve

         prob = model.predict_proba(x_train)[:,1]
         fpr, tpr, thresh = roc_curve(train['spam'], prob, pos_label=1)
         plt.step(fpr, tpr)
         plt.xlabel('False Positive Rate')
         plt.ylabel('True Positive Rate')
         plt.title('ROC Curve')
```

```
Out[51]: Text(0.5, 1.0, 'ROC Curve')
```


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