
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:

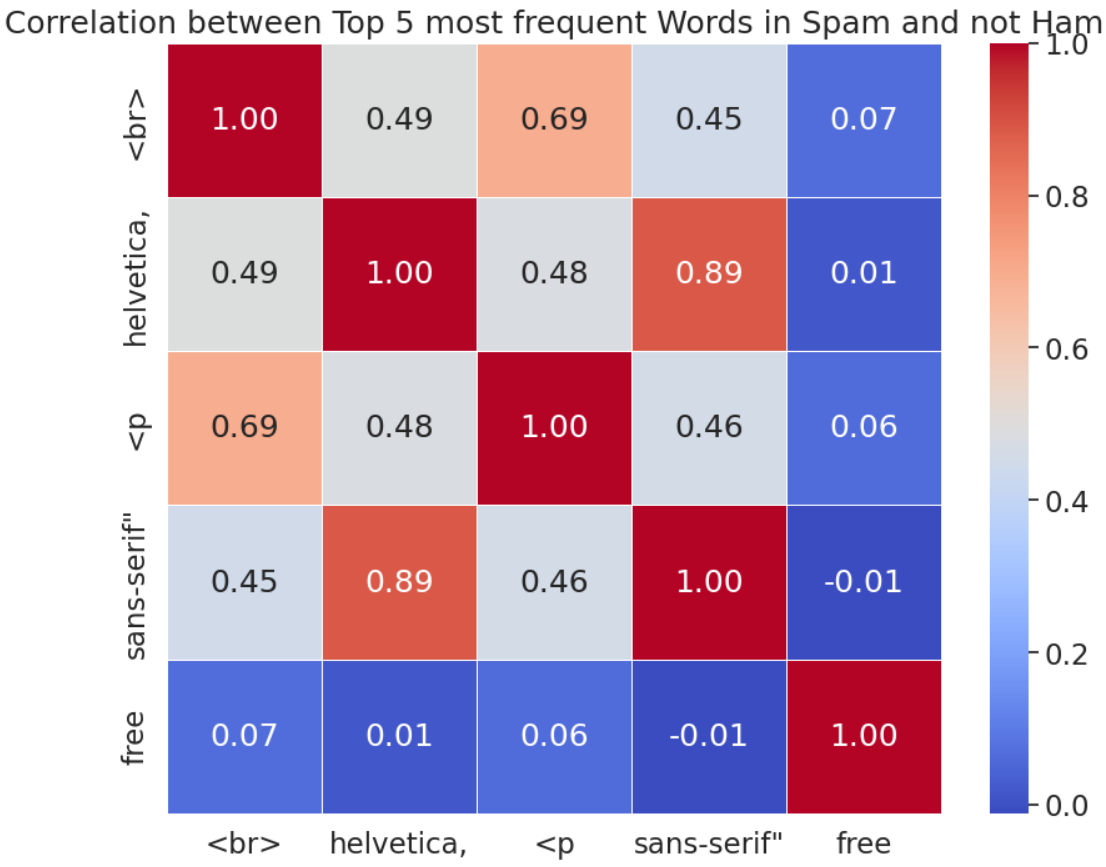
1. How did you find better features for your model?
2. What did you try that worked or didn't work?
3. What was surprising in your search for good features?

1. To find better features I followed the guidance to find better words to use as features. I decided that the best way to do so would be to choose words that appear in Spam and don't appear in Ham to best distinguish both emails. I then created two dictionaries with the words in both ham and spam emails. Then I sorted the words to find the 130 most frequent words in each dictionary and created a list with the most frequent words appearing in spam that do not appear in ham. This allowed me to select these words as the best features for my model.
2. I first tried to increase the number of words being sorted but ran into the error of reaching the max number of iterations. Additionally, I tried looking at different possible features in the email text, as suggested, such as the number of capital letters in spam and ham, and found that there was no significant difference in using it as a feature. Additionally, I looked at the difference in punctuation between spam and ham emails and found that although it appeared to be a good feature, choosing the words that most commonly appear in Spam and not in Ham seemed to be the best feature to use due to the distinctiveness we can obtain.
3. It surprised me that when looking at the words that were the best features, it was those that weren't actually words but coinages of letters and characters that seemed to be the best features. This was the case for '', for example, and the font name Helvetica which were the two words that appeared the most in spam and did not appear in ham. It also surprised me how simple punctuation marks could be used to better distinguish between spam and ham emails

0.2 Question 2a

Generate your visualization in the cell below.

```
In [60]: words_in_spam5 = words_in_spam[0:5]
         word_features = words_in_texts(words_in_spam5, original_training_data['email'])
         word_features_df = pd.DataFrame(word_features, columns=words_in_spam5)
         data_with_word_features = pd.concat([original_training_data, word_features_df], axis=1)
         spam_data = data_with_word_features[data_with_word_features['spam'] == 1]
         correlation_matrix = spam_data[words_in_spam5].corr()
         plt.figure(figsize=(10, 8))
         sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)
         plt.title('Correlation between Top 5 most frequent Words in Spam and not Ham')
         plt.show()
```



0.3 Question 2b

Write your commentary in the cell below.

The heatmap portrays the correlation between word features I considered choosing for my model. I decided to observe the correlation between these words as the plot will tell me the likelihood that if one word appears in spam emails the likelihood of the other word appearing will also be high. An example is 'helvetica' and 'san-serrif' which show a significant positive correlation suggesting the predominant presence of font type names in such spam emails, unlike 'sans-serif' and 'free' which show a negative correlation. This helped me choose which potential words I wanted to exclude or include as features and confirm that I could justify using them as features for my model as they so show correlation indeed

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.

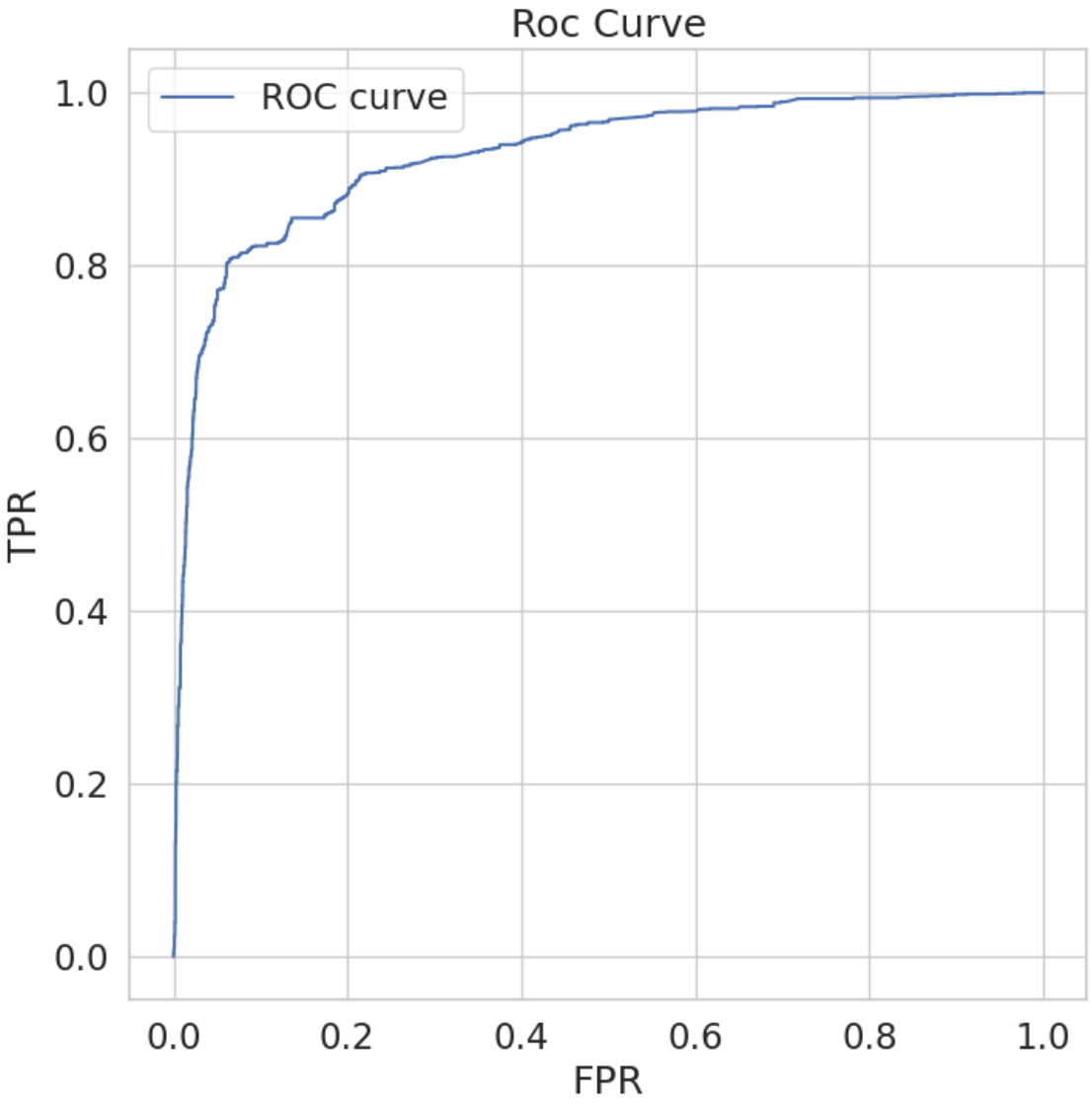
Hint: You'll want to use the `.predict_proba` method for your classifier instead of `.predict` to get probabilities instead of binary predictions.

In [68]:
```
from sklearn.metrics import roc_curve

y_predict = my_model.predict_proba(x_train)[:,1]
false_postive,true_positive,thresholds = roc_curve(y_train,y_predict)

plt.figure(figsize=(8, 8))
plt.plot(false_postive, true_positive, label='ROC curve')

plt.xlabel('FPR')
plt.ylabel('TPR')
plt.title('Roc Curve')
plt.legend()
plt.show()
```


Roc Curve
ROC curve
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0