# Notebook

April 26, 2020

### 0.0.1 Question 6c

Provide brief explanations of the results from 6a and 6b. Explain why the number of false positives, number of false negatives, accuracy, and recall all turned out the way they did.

The model always predicts negative, so we can't have any false positives. The number of false negatives will simply be the number of true positives in the training set. Our accuracy is the percent of correct responses, which is simply the number of negatives out of the total number of instances (the same as 1 minus the number of positives out of the number in the whole set). Recall, which has false positive in the numerator, will subsequently become 0.

### 0.0.2 Question 6e

Are there more false positives or false negatives when using the logistic regression classifier from Part A?

More false negatives (1699) than false positives (122)

### 0.0.3 Question 6f

1. Our logistic regression classifier got 75.8% prediction accuracy (number of correct predictions / total). How does this compare with predicting 0 for every email?
2. Given the word features we gave you above, name one reason this classifier is performing poorly. Hint: Think about how prevalent these words are in the email set.
3. Which of these two classifiers would you prefer for a spam filter and why? Describe your reasoning and relate it to at least one of the evaluation metrics you have computed so far.

1. The zero predictor has an accuracy of 74.5, so our logistic classifier isn't much better.
2. The words most likely aren't particularly prevalent in the email set, and so they aren't particularly useful for our model.
3. I would still prefer the logistic predictor. The zero predictor simply assumes that nothing is spam, and so it would be completely useless for attempting to filter out spam. The precision and recall rate of the zero predictor are both 0.

### 0.0.4 Question 7: Feature/Model Selection Process

In the 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 / didn't work?
3. What was surprising in your search for good features?

I wanted to see how many words were necessary to reach the desired accuracy, so I didn't seek out different features for my model. Interestingly enough, what worked was just adding more and more words. Adding 1 or 2 words at a time would decrease my accuracy, but adding 5 words at once almost always increased my accuracy. By combing through the raw data for words that seemed to be distinctly "spammy" or "hammy", I was able to come up with a list of 40 words that provides me with a 90% accuracy on the training set.

Generate your visualization in the cell below and provide your description in a comment.

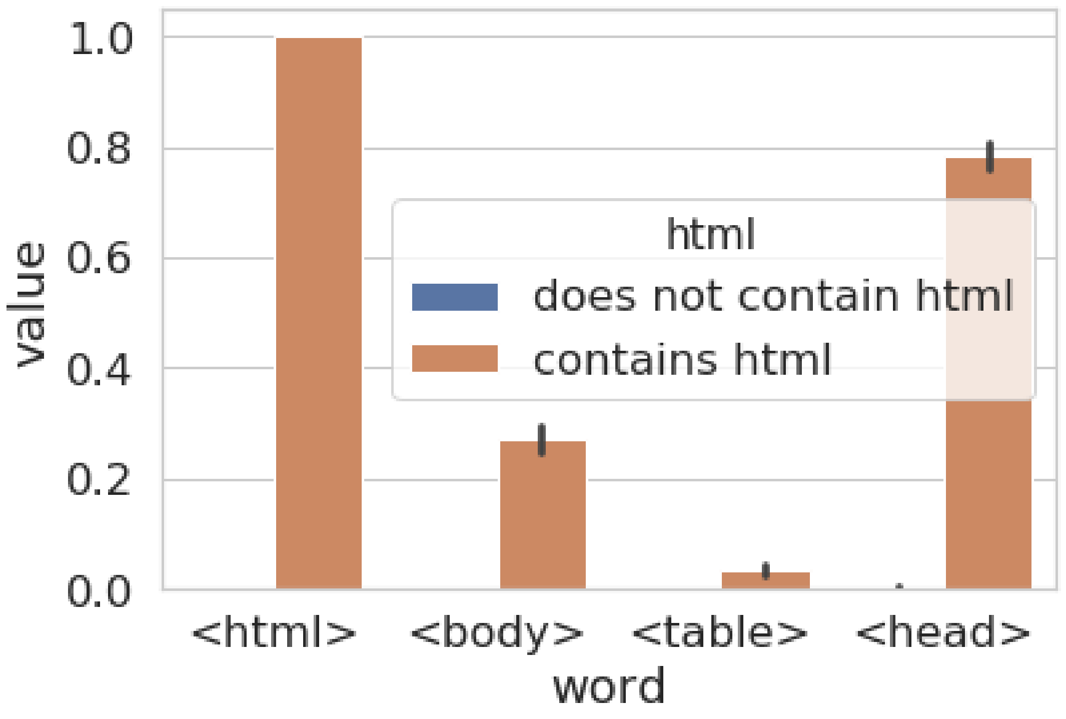
```
In [159]: # Write your description (2-3 sentences) as a comment here:
          # I promise it's not a basic barchart from question 3. I noticed that some of the emails have
          # multiple words I could use to parse for this - <html>, <body>, <table>, and <head> to name
          # I needed to include each of these, or if simply <html> would suffice for training my model.
          # the data for each of these words, and for

          # Write the code to generate your visualization here:

          html = train.email.str.contains('<html>')
          body = train.email.str.contains('<body>')
          table = train.email.str.contains('<table>')
          head = train.email.str.contains('<head>')

          d = {'<html>': 1*np.array(html).T, '<body>': 1*np.array(body).T,
               '<table>': 1*np.array(table).T, '<head>': 1*np.array(head).T}
          email_pd = pd.DataFrame(data=d)
          email_pd['html'] = ['contains html' if type == True else 'does not contain html' for type in
          email_pd = email_pd.melt('html')
          email_pd = email_pd.rename(columns={'variable': 'word'})
          sns.barplot(data=email_pd, x='word', y='value', hue='html')
          plt.show()

          # Note: if your plot doesn't appear in the PDF, you should try uncommenting the following lin
          # plt.show()
```


value
html
does not contain html
contains html
<html>
<body>
<table>
<head>
word