Notebook

April 18, 2020

0.0.1 Question 1c

Discuss one thing you notice that is different between the two emails that might relate to the identification of spam.

The spam email seems to be written in html while the ham email is not and seems to be just a normal string.

0.0.2 Question 3a

Create a bar chart like the one above comparing the proportion of spam and ham emails containing certain words. Choose a set of words that are different from the ones above, but also have different proportions for the two classes. Make sure to only consider emails from `train`.

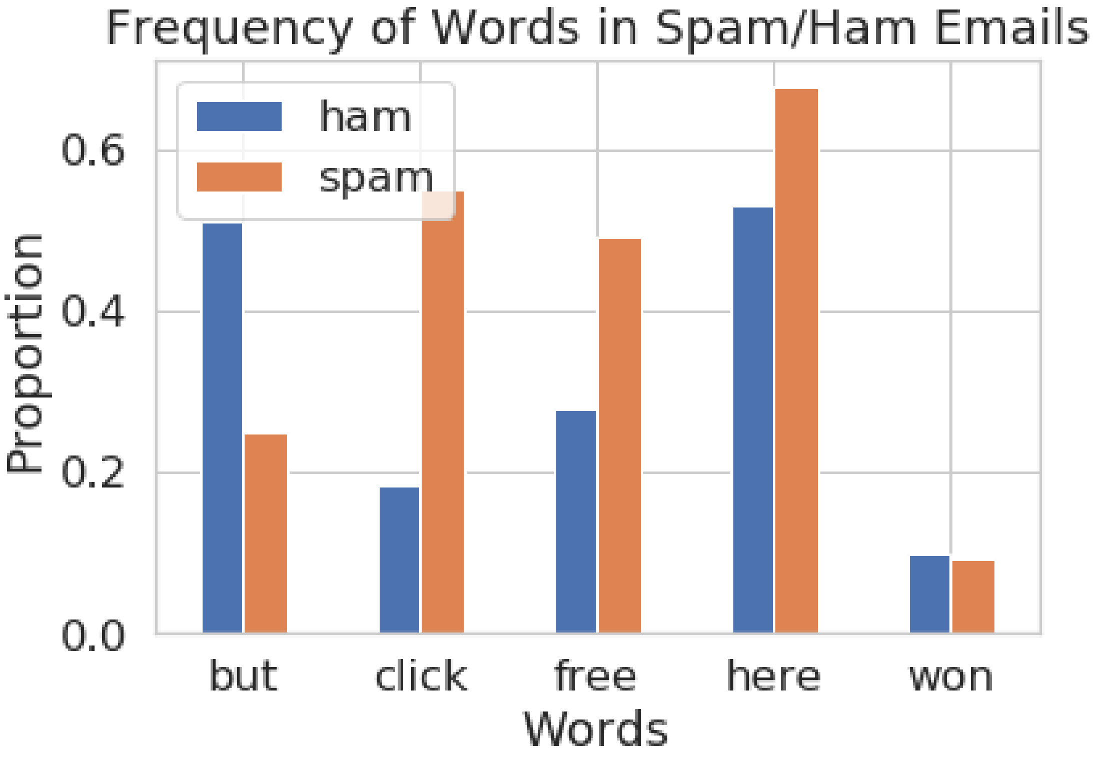
```
In [33]: train=train.reset_index(drop=True) # We must do this in order to preserve the ordering of emai

         num_spam = train['spam'].sum()
         num_ham = len(train['spam']) - train['spam'].sum()

         def convert_spam_ham(series):
             new_array = []
             for x in series:
                 if x == 0:
                     new_array.append('ham')
                 else:
                     new_array.append('spam')
             return new_array

         spam_indicator_text = ['won', 'but', 'click', 'free', 'here']

         text = train['email']
         indicator_array = words_in_texts(spam_indicator_text, text)
         df = pd.DataFrame(indicator_array, columns = spam_indicator_text).assign(type = convert_spam_ha
         df_melted = df.melt('type')
         df_melted_sorted = df_melted.groupby(['variable', 'type']).sum().unstack()
         df_melted_sorted = df_melted_sorted['value'].assign(ham = df_melted_sorted['value']['ham'] / nu
                                         spam = df_melted_sorted['value']['spam']/num_spam)
         df_melted_sorted.plot(kind = 'bar')
         plt.legend(['ham', 'spam'])
         plt.xticks(rotation=0)
         plt.xlabel('Words')
         plt.ylabel('Proportion')
         plt.title('Frequency of Words in Spam/Ham Emails')
         plt.show();
```

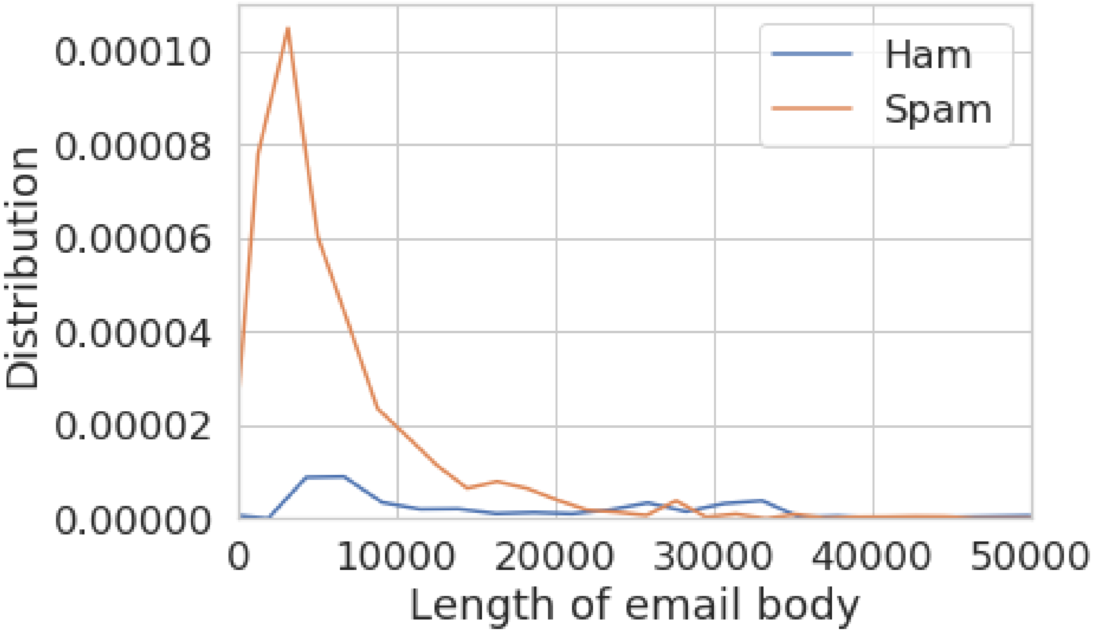

Frequency of Words in Spam/Ham Emails
ham
spam
Proportion
0.0
0.2
0.4
0.6
but
click
free
here
won
Words

0.0.3 Question 3b

Create a *class conditional density plot* like the one above (using `sns.distplot`), comparing the distribution of the length of spam emails to the distribution of the length of ham emails in the training set. Set the x-axis limit from 0 to 50000.

```
In [34]: spam_email_lengths = pd.DataFrame({'spam':train[train['spam'] == 1]['email'].apply(len)})
         ham_email_lengths = pd.DataFrame({'ham':train[train['spam'] == 0]['email'].apply(len)})

         sns.distplot(ham_email_lengths, hist = False, label = 'Ham')
         sns.distplot(spam_email_lengths, hist = False, label = 'Spam')
         plt.xlabel('Length of email body')
         plt.ylabel('Distribution')
         plt.xlim(0, 50000)
         plt.show();
```


Ham
Spam
Distribution
Length of email body