Notebook

April 27, 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.

Notice that we are using a 'zero_predictor' classifier which means that this specified classifier will predict every single email in a dataset as 'ham' emails; thus, will never predict a email as spam. This means that our classifier will never have 'True Positives'; therefore, the recall of the 'zero_predictor' will always be 0 regardless of the dataset its working with. The number of false positives is 0 because the 'zero_predictor' will never mislabel an email as spam (regardless of dataset) since it only predicts/labels emails as ham. The number of false negatives is 1918 since the 'zero_predictor' will inevitably mislabel an email as ham when it is truly spam (of course with the condition that the classifier is working on a diversified dataset that consist of both spam and ham emails). As for the 'zero_predictor' classier's accuaracy, it is about 74.47% accurate which makes sense since the 'zero_predictor' classifier will inevitably make correct predictions under the condition that the dataset is diversified with spam and ham emails. In this case, the accuracy being high idicates that in the ('training') dataset there is a bigger proportion of ham emails as opposed to spam emails.

0.0.2 Question 6e

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

When using the logistic regression classifier from Part A, there are more 'False Negatives' than 'False Positives'. There is 1821 'False Negatives' vs 122 'False Positives'.

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. Our logistic regression classifier prediction accuracy (of 75.8%) does slightly better than the 'zero_predictor' classifier predicition accuaracy (of 74.47%) because now the new logistic regression classifier is no longer predicting just ham emails for every single email. In other words, the new logistic regression classifier is predicting both ham and spam emails depending on the features/characteristics of the given email; therefore, the increase of prediction accuracy (from the original zero_predictor classifier) is due to the new ability of the classifier to predict both ham and spam emails.

2. One reason this classifier is performing poorly is due to the word features choosen for training the classifiers accurate prediction abilities. For instance, the words that most likely results to the classifiers poor performance are 'drug', 'bank', 'prescription', 'memo', and 'private' because of their minimal prevalence in emails. For instance, the words 'drug', 'bank', 'prescription', 'memo', and 'private' respectively have the following percentage of prevalence in emails: 1.623852%, 4.07294% , 0.732064%, 4.126181%, 4.099561%. The percentage of prevalence in emails (just stated) are extremely low which contributes to the poor performance of the logistic regression classifier.

3. Of these two classifiers, I would prefer the zero_predictor classifier since when using this classifier there is a zero chance of the classifier mislabeling/mispredicting a ham email as spam (i.e. it has a false alarm rate of 0). This could potentially be an important factor to me if I used emails solely for business relations and I accidently never read an important email/emails at a timely manner (or at all) due to the classifier mislabeling/mispredicting important ham email(s) as spam. For some individuals, the tradeoff between having a classifier with higher prediction accuaracy (the logisitic regression classifier) and a 2.18% false alarm rate is acceptable. However, personally I would perfer the classifier with a slightly lower prediction accuracy (the zero_predictor classifier) and a zero false alarm rate.

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?

1. First, I examined the frequency of each word that was of length four or more within both the ham and spam email (note: other symbols were removed prior from exmaination). I created two seperate dataframes for ham and spam email that contained a column of words frequency. After, I saved the top 35 most frequent words for the spam and ham email. Also, I examined the frequency of each word that was of length two or more within both the ham and spam subject area of the email. Then, as before, I created two seperate dataframes that contained a column of the different word frequencies for ham and spam subject section of the email. Last, as before, I saved the top 35 most frequent words for both the spam and ham email subject section. Another way I improved the features for my model was that I inspected the subject section of both the ham and spam emails to see how frequency the symbols "$" and "!". It was conclude that these symbols were more frequently used among the spam emails. After, I created a wordcloud visualization in order to recognize which words had the highest frequency (and most importance) in both spam and ham emails. All this helped with finding the best features (i.e. words) for my model in order to optimize percision.

2. Plotting the frequency of each word for both ham and spam email/subject section worked effectively to supplement words I found in the wordcloud. However, plotting the lengths of emails and then using the information for my model decreased my validation score.

3. Something that took me by surprised was how using the length of emails as a feature caused negative effect to my model's validation score. I anticipated that including the emails length as a feature would extremely boost my validation score and increase the accuracy of my model; however, I was wrong. Also, I found surprising that using symbols as features increased my validation score to a higher degree than if I were to use solely the words I found using my barplots and WordCloud visualt.

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

In [898]:
```
# Write your description (2-3 sentences) as a comment here:
# The following visualization is a WordCloud that depicts words that were frequently seen amo
# Words, on the visualization, that consist of bigger and bolder fonts depict words that were
# hence important. Many of the words depicted in the following visualization helped me streng
#(as they were used as features for my model).

# Write the code to generate your visualization here:
stopwords = set(STOPWORDS)

wordcloud = WordCloud(width = 800, height = 800, background_color = 'white', stopwords = stopw

plt.figure(figsize = (8,8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
#Note: if your plot doesn't appear in the PDF, you should try uncommenting the following line
plt.show()
```