Data 100/200 Homework 9 Written

Jackie Hu

TOTAL POINTS

9.5 / 10

QUESTION 1

1 Question 1 **2 / 2**

✓ **+ 2 pts States a difference between the two types of emails**

+ **0 pts** Incorrect/blank

QUESTION 2

2 Question 3 **2 / 2**

✓ **+ 2 pts Correct plot and labels**

+ **1 pts** Missing axes labels, titles, or key
+ **1 pts** Missing or incorrect plot
+ **0 pts** Incorrect/Blank

QUESTION 3

3 Question 6c **2 / 2**

✓ **+ 2 pts Correct**

+ **1 pts** Only mentions up to 3 of FP, FN, accuracy and recall (not all 4)
+ **0 pts** Incorrect/blank

QUESTION 4

4 Question 6e **1 / 1**

✓ **+ 1 pts Correct: There are more false negatives (FN = 1699) than false positives (FP = 122).**

+ **0 pts** Incorrect/Blank

QUESTION 5

5 Question 6f **2.5 / 3**

✓ **+ 1 pts Part 1 correct - both models have similar accuracy**

+ **1 pts** Part 2 correct - low prevalence, poor choice of words, or poor word differentiation between spam and ham emails

✓ **+ 1 pts Part 3 correct - thoughtful response referencing one or more evaluation metrics**

+ **0 pts** Blank or completely wrong

+ 0.5 Point adjustment

The words that we've chosen as features are not prevalent, thus X_train is very sparse.

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

- ham: shorter comparing to the spam, specific person's name is mentioned, which feels more personalized formatting: more like a message style, there's no html tags
- spam key words: garanteed, increase size, get the job done come in here and see how; that said lots of action incentive words. formatting: html tags, url embeded in the message

0.0.1 Question 3

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

```
Out[60]:         id                                            subject  \
         0     7657             Subject: Patch to enable/disable log\n
         1     6911        Subject: When an engineer flaps his wings\n
         2     6074  Subject: Re: [Razor-users] razor plugins for m…
         3     4376  Subject: NYTimes.com Article: Stop Those Press…
         4     5766  Subject: What's facing FBI's new CIO? (Tech Up…
         …      …                                               …
         7508  5734  Subject: [Spambayes] understanding high false …
         7509  5191        Subject: Reach millions on the internet!!\n
         7510  5390                      Subject: Facts about sex.\n
         7511   860  Subject: Re: Zoot apt/openssh & new DVD playin…
         7512  7270  Subject: Re: Internet radio - example from a c…

                                                           email  spam
         0     while i was playing with the past issues, it a…     0
         1     url: http://diveintomark.org/archives/2002/10/…     0
         2     no, please post a link!\n \n fox\n ----- origi…     0
         3     this article from nytimes.com \n has been sent…     0
         4     <html>\n <head>\n <title>tech update today</ti…     0
         …                                                   …   …
         7508  >>>>> "tp" == tim peters <tim.one@comcast.net>…     0
         7509  \n dear consumers, increase your business sale…     1
         7510  \n forwarded-by: flower\n \n did you know that…     0
         7511  on tue, oct 08, 2002 at 04:36:13pm +0200, matt…     0
         7512  chris haun wrote:\n > \n > we would need someo…     0

         [7513 rows x 4 columns]
```

```
In [61]: #choosing set of spam keywords
         words_set = ['guaranteed', '$', 'increase', 'size', 'totally', 'price', 'action']

         #get the email contents as pd series
         ham_emails_contents = train[train['spam'] == 0]['email']
         spam_emails_contents = train[train['spam'] == 1]['email']

         #out put the matrix form
         words_in_ham = words_in_texts(words_set, ham_emails_contents)
         words_in_spam = words_in_texts(words_set, spam_emails_contents)

         #count the amount of words in each category using np.sum
```

```
words_in_ham_cnt = np.sum(words_in_ham, axis=0)
words_in_spam_cnt = np.sum(words_in_spam, axis=0)
```

In [71]:
```
words_in_spam_cnt
```

Out[71]:
```
array([229, 758, 188, 959,  49, 333, 425])
```

In [77]:
```
#visualize in df
df_spam_cnt = pd.DataFrame({'Words': words_set, 'Count_spam': words_in_spam_cnt.tolist()})
df_spam_cnt
```

Out[77]:

	Words	Count_spam
0	guaranteed	229
1	$	758
2	increase	188
3	size	959
4	totally	49
5	price	333
6	action	425

In [63]:
```
words_in_ham_cnt
```

Out[63]:
```
array([ 21, 694, 153, 493,  70, 300, 477])
```

In [76]:
```
#visualize in df
df_ham_cnt = pd.DataFrame({'Words': words_set, 'Count_ham': words_in_ham_cnt.tolist()})
df_ham_cnt
```

Out[76]:

	Words	Count_ham
0	guaranteed	21
1	$	694
2	increase	153
3	size	493
4	totally	70
5	price	300
6	action	477

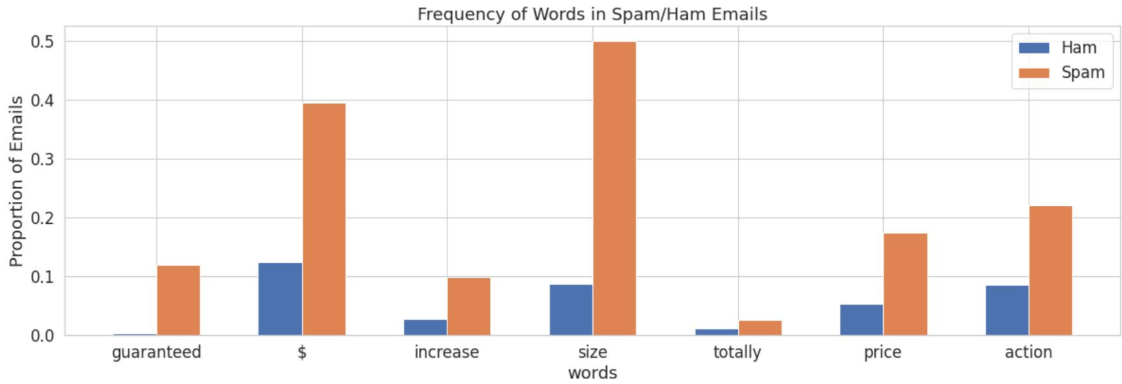
In [69]:
```
#visulization
bar_width = 0.3
ham_total = len(ham_emails_contents)
spam_total = len(spam_emails_contents)
```

```
ham_proportion_height = words_in_ham_cnt/ham_total
spam_proportion_height = words_in_spam_cnt/spam_total

#plotting the barchart
fig= plt.figure(figsize=(20,6))
plt.bar(x = words_set, align='edge', height = ham_proportion_height, label='Ham', width = -bar_
plt.bar(x = words_set, align='edge', height = spam_proportion_height, label='Spam', width = ba

plt.legend()
plt.xlabel('words')
plt.ylabel('Proportion of Emails')
plt.title('Frequency of Words in Spam/Ham Emails')

plt.show()
```



0.0.2 Question 6c

Provide brief explanations of the results from 6a and 6b. Why do we observe each of these values (FP, FN, accuracy, recall)?

- if we predict 0 on all cases, we would reach an accuracy score of almost 74.5%; which makes sense sinve we normally have less spam then ham, and predict all emails as ham would till have most of the email labeled correctly.
- however we have 0 on the recall, because we are not labelling any spam correctly, therefore the false positive score is 0

0.0.3 Why

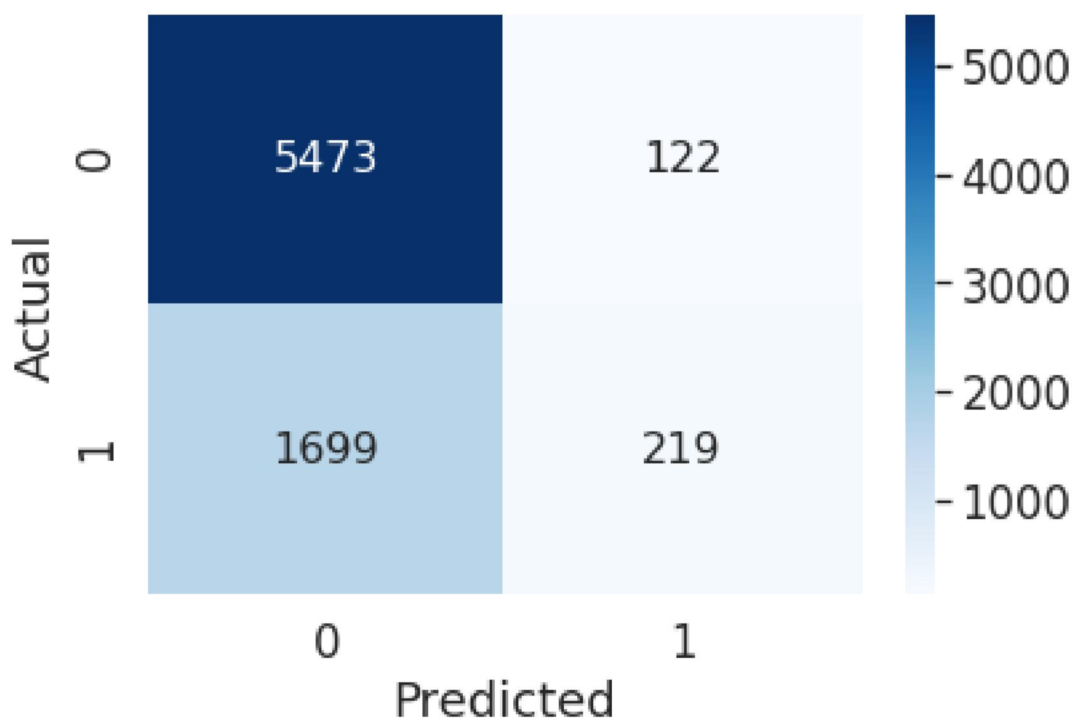
- False positive (FP):: the number of ham emails that are mislabeled as spam and filtered out of the inbox
- False negative (FN): the number of spam that are mislabeled as hams.
- Accuracy/precision: to know how many hams are labeled correctly out of the total number of emails.
- Recall is for us to know how many spams are correctly labeled as spam, since it's zero predictor, so no spam is labeled.

0.0.4 Question 6e

Are there more false positives or false negatives when using the logistic regression classifier from Question 5?

In [94]:
```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_train, Y_pred)

#plot confusion matrix heatmap
sns.heatmap(cm, annot=True, fmt = 'd', cmap = 'Blues', annot_kws = {'size': 16})
plt.xlabel('Predicted')
plt.ylabel('Actual');
```



In [112]:
```
logistic_predictor_far
```

Out[112]: 0.021805183199285077

Based on the confusion matrix, the false_positive has value of 122, false_negative has a value of 1699; there are more false negatives than false positives

0.0.5 Question 6f

1. Our logistic regression classifier got 75.76% 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.

- Always predicting 0 has an accuracy score of 74.47%, our model has an accuracy score of 75.76%, therefore our logistic regression classifier is lightly better than the random predictor.
- Seems like there're words are common words, and exists in both spam and ham emails, which might not be the good indicator for classification.
- Depends on the circumstances, normally I'd prefer the logistic regression model classifier for a spam filter, because it has higher prediction accuracy. But for my berkeley school email address I'd prefer the 0 classifier, because I don't want to miss any email, since the False-alarm rate is 0 for that classifer, I'd use the 0 classifier for the berkeley email inbox.