
0.1 Question 1

Discuss one attribute or characteristic you notice that is different between the two emails that may allow you to uniquely identify a spam email.

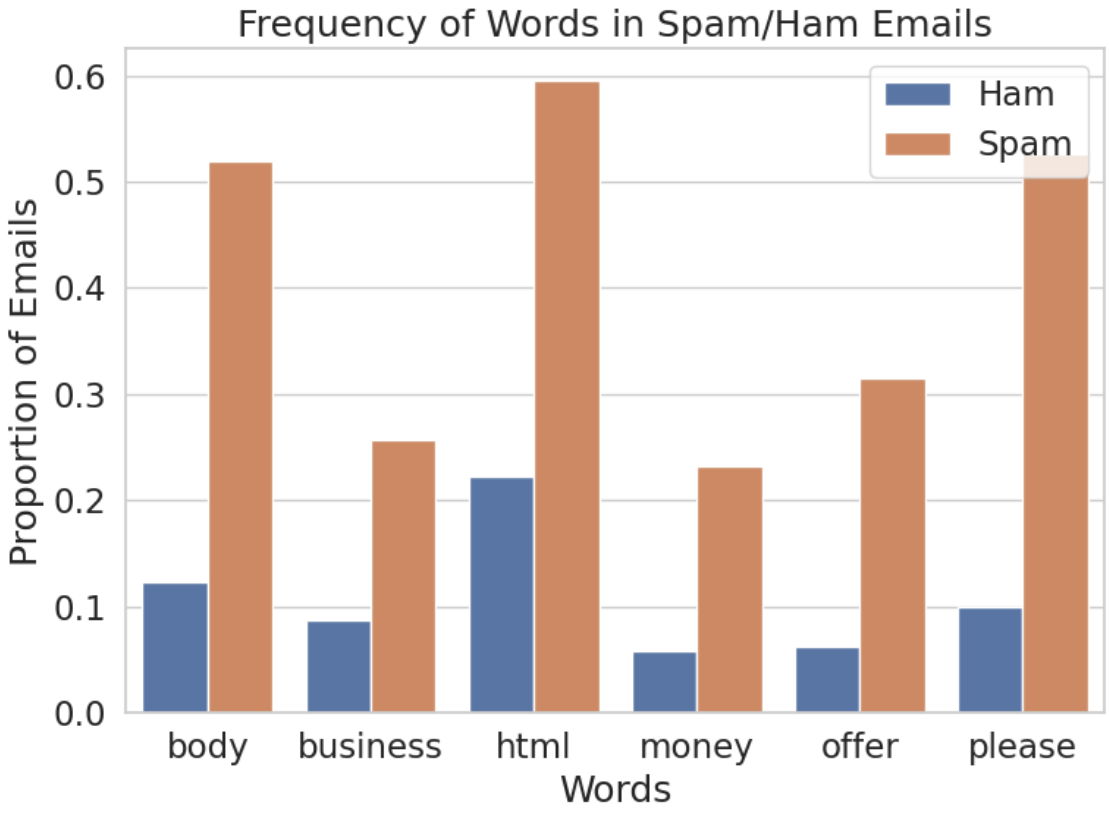
Spam mail was written in html email. Spam mail might have written in html format because they wanted to decorate their emails to catch people's eyes.

Create your bar chart in the following cell:

```
In [13]: train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em
         arr = words_in_texts(["body", "business", "html", "money", "offer", "please"], train["email"])
         df_arr = pd.DataFrame(arr)
         df_arr = df_arr.rename(columns = {0: "body", 1: "business", 2: "html", 3: "money", 4: "offer",
         df_arr_with_type = pd.concat([df_arr, train["spam"]], axis = 1)
         df_arr_spam = df_arr_with_type[df_arr_with_type["spam"] == 1]
         df_arr_ham = df_arr_with_type[df_arr_with_type["spam"] == 0]
         spam_prob = df_arr_spam.iloc[:, 0:6].sum(axis = 0) / df_arr_spam.shape[0]
         ham_prob = df_arr_ham.iloc[:, 0:6].sum(axis = 0) / df_arr_ham.shape[0]
         df = pd.concat([pd.DataFrame(ham_prob).T, pd.DataFrame(spam_prob).T], axis = 0)
         df = df.reset_index()
         df.iloc[0, 0] = "Ham"
         df.iloc[1, 0] = "Spam"
         df = df.melt("index")

         plt.figure(figsize=(8,6))
         sns.barplot(x = "variable", y = "value", hue = "index", data = df)
         plt.title("Frequency of Words in Spam/Ham Emails")
         plt.xlabel("Words")
         plt.ylabel("Proportion of Emails")
         plt.gca().legend().set_title('')


         plt.tight_layout()
         plt.show()
```


Frequency of Words in Spam/Ham Emails
Ham
Spam
Proportion of Emails
0.0
0.1
0.2
0.3
0.4
0.5
0.6
body
business
html
money
offer
please
Words

0.2 Question 6c

Explain your results in `q6a` and `q6b`. How did you know what to assign to `zero_predictor_fp`, `zero_predictor_fn`, `zero_predictor_acc`, and `zero_predictor_recall`?

Since the zero predictor always predicts 0 (ham), it will never predict 1, meaning there are no false positives (zero_predictor_fp = 0). However, for every email that is actually spam (label 1), the zero predictor incorrectly predicts ham, creating a false negative for each spam email. That's why zero_predictor_fn equals the total number of spam emails, which is sum(Y_train == 1).

For accuracy, the zero predictor correctly classifies all ham emails (label 0). Therefore, its accuracy is the proportion of emails that are ham: sum(Y_train == 0) / len(Y_train). For recall, since it never correctly identifies any spam (no true positives), the recall is 0.

0.3 Question 6f

How does the accuracy of the logistic regression classifier `my_model` compare to the accuracy of the zero predictor?

Our logistic regression classifier had an accuracy of approximately 75.76%, while the zero predictor had an accuracy of 74.47%. Although the logistic regression model performs slightly better, the improvement is relatively small. This indicates that even though logistic regression is learning from the data, the features provided are not strong enough to create a highly accurate classifier beyond simply always predicting ham (0).

0.4 Question 6g

Given the word features provided in Question 4, discuss why the logistic regression classifier `my_model` may be performing poorly.

Hint: Think about how prevalent these words are in the email set.

The logistic regression classifier may be performing poorly because the selected words were not strong indicators to separate spam from ham. Some words, like "prescription", appear extremely rarely (in less than 1% of emails), making them weak signals. Other words like "drug", "bank", and "private" might occur in both spam and non-spam contexts, reducing their usefulness. In general, the feature set was too small and not representative enough of typical spam characteristics. A better-performing classifier would require a larger set of more common and discriminative words to distinguish spam from ham more effectively. Additionally, a more diverse and richer training dataset would help improve model performance.

0.5 Question 6h

Would you prefer to use the logistic regression classifier `my_model` or the zero predictor classifier for a spam filter? Why? Describe your reasoning and relate it to at least one of the evaluation metrics you have computed so far.

I would prefer to use the zero predictor in this case. Although the logistic regression classifier detects more spam emails (higher recall), it also produces more false positives, meaning it mistakenly flags legitimate emails as spam. In real-world email filtering, false positives are highly damaging because users might miss important emails. The zero predictor, while failing to detect spam, keeps the false positive rate extremely low, ensuring that important ham emails are safely delivered. Since avoiding false positives is critical, I would prefer the zero predictor for this specific situation, based on its lower false positive rate.