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

The tone of the ham email is simply to inform the user of the auction where as the spam seems more like an ad. It starts by appealing to mascuinity and tries to make the reader insecure about their size, then it baits them in with a “guarantee” that you can increase your size. Identifying phrases or words that indicate promise or a sense of urgency likely has a high chance to be spam which can be used to indentify it.

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 [27]: train_words = ['feedback','spending','membership','increase', 'result', 'yes']
         #train_words = ['body','business','html','money', 'offer', 'please']
```

```
In [28]: train = train.reset_index(drop=True) # We must do this in order to preserve the ordering of em
         check = words_in_texts(train_words, train['email'])
         spam_number = len(train[train['spam'] == 1])
         ham_number = len(train[train['spam'] == 0])
         words_df = pd.DataFrame(check)
         spam_or_ham = train['spam'].replace({0:'ham', 1:'spam'})
         words_df['type'] = spam_or_ham
         words_df = words_df.rename({0:'feedback',1:'consumers',2:'membership',3:'increase',4:'result',5
         #words_df = words_df.rename({0:'body',1:'buisness',2:'html',3:'money',4:'offer',5:'please'},ax

         words_df = words_df.melt(id_vars = "type")
         words_df = words_df.groupby(["variable","type"]).sum()
         words_df = words_df.reset_index()
         words_df["proportion"] = words_df["value"]
         proportions = words_df["proportion"].to_list()
         for i in range(len(proportions)):
             if i % 2 == 0:
                 proportions[i] = proportions[i] / ham_number
             else:
                 proportions[i] = proportions[i] / spam_number
         words_df["proportion"] = proportions
         words_df
```

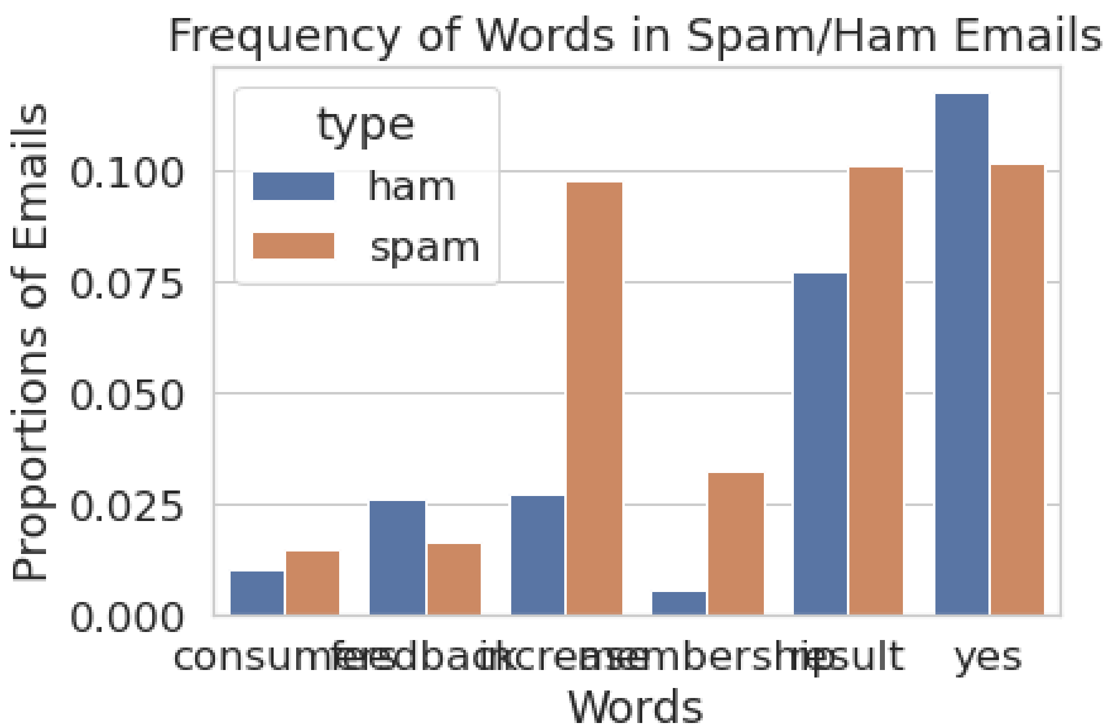
Out[28]:

	variable	type	value	proportion
0	consumers	ham	58	0.010366
1	consumers	spam	29	0.015120
2	feedback	ham	148	0.026452
3	feedback	spam	32	0.016684
4	increase	ham	153	0.027346
5	increase	spam	188	0.098019
6	membership	ham	34	0.006077
7	membership	spam	62	0.032325
8	result	ham	435	0.077748
9	result	spam	195	0.101668
10	yes	ham	659	0.117784
11	yes	spam	196	0.102190

```
In [29]: sns.barplot(x='variable',y='proportion',hue = 'type', data = words_df)
         plt.xlabel("Words")
```

```
plt.ylabel("Proportions of Emails")
plt.title("Frequency of Words in Spam/Ham Emails")
plt.figure(figsize = [2,1])
```

Out[29]: <Figure size 144x72 with 0 Axes>



<Figure size 144x72 with 0 Axes>

0.0.2 Question 6c

Comment on the results from 6a and 6b. For **each** of FP, FN, accuracy, and recall, briefly explain why we see the result that we do.

For 6a there cannot be any false positives since our zero_predictor always flags mail as ham. On the opposite side because of that the number of false negatives would be the number of spam emails since our predictor calls everything ham. In 6b since there are no false positives the precision will always be 0 since there will be zero true positives. For recall the same applies as there are no true positives since everyhing returns as negative/0.

0.0.3 Question 6e

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

There are more false positives because before with the zero classifier it always predicted something was not spam so hams could never be marked as spam. Withe the logistic regression classifier it now has the non-zero chance to classify some ham emails as spam.

0.0.4 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.

In [40]:
```
1 - len(train[train['spam'] == 1]) / len(train)
```

Out[40]:
```
0.7447091707706641
```

1. Predicting 0 gives us about a 74.47% accuaracy which is worse than the logistic classifier.
2. A reason that our current classifier may be underperforming would be that words that were chosen did not appear in enough of the emails making it so that the proportions were skewed because a large enough sample size of emails was not chosen.
3. I would prefer the logistic regression filter because it is more accurate and it can actually filter emails, where as the zero classifier basically is not even a filter.