A list of the top 10 bag of word features selected by filtering by frequency.
['to', 'you', 'I', 'a', 'the', 'and', 'is', 'in', 'i', 'u']

A list of the top 10 bag of word features selected by filtering by mutual information.
['Call', 'to', 'call', 'or', 'FREE', 'claim', 'To', 'mobile', '&', 'Txt']

Run logistic regression with the top 10 words by frequency. Compare the accuracy of the model learned with 10 most frequent words to the model that predicts the most common class. Increase the number of features selected by 5 until you outperform the most common class model. What number do you need?
MostCommonClass Model: 84.10%
I would need 25 features before a frequently occurring word provided indication of spam/not spam. Until then, the model simply labelled everything as not spam just like MostCommonClass model. This is probably because there is a lot of entropy in the most commonly occurring english words such as “is”.

Run logistic regression with the top 10 words by mutual information. Produce a table showing the selected words and the weights learned for them.
w0: -3.099245 w1: 2.364639 w2: 1.280651 w3: 1.877932 w4: 1.572349 w5: 1.572342 w6: 1.267719 w7: 1.486217 w8: 1.353684 w9: 1.663527 w10: 1.609621

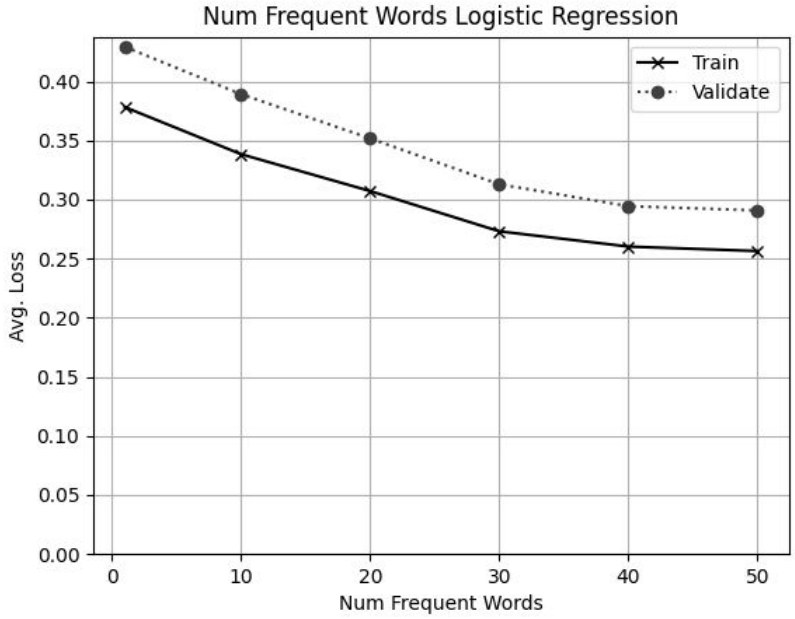
Word	Weight
claim	2.364639
won	1.280651
prize	1.877932
FREE	1.572349
awarded	1.572342
URGENT!	1.267719
PO	1.486217
&1000	1.353684
selected	1.663527
Box	1.609621

Create an if statement that partially matches the linear model, classifying some of the same messages as spam (with no additional false positives). Use no more than 5 clauses in the if statement.
e.g. If has_word(X) and has_word(Y) then classify as spam

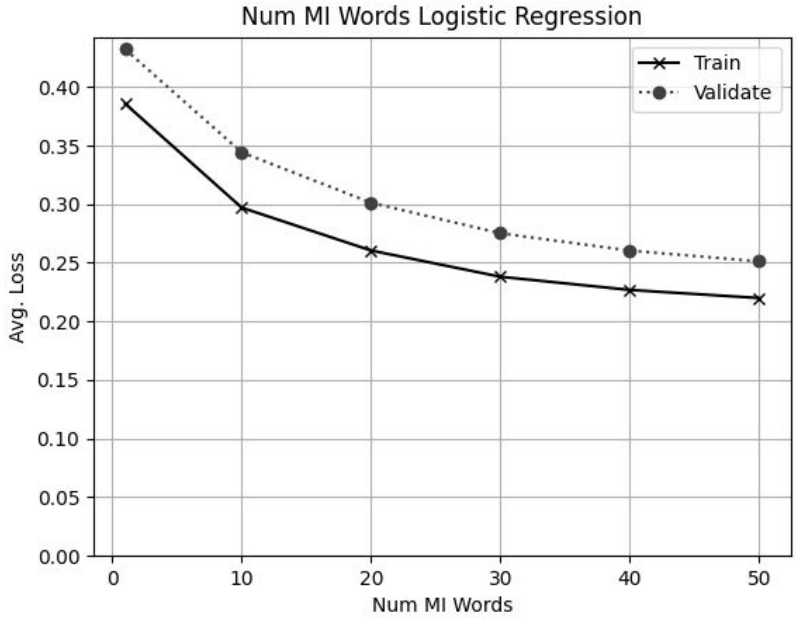
if (has_word(“claim”) and has_word(“)) or (has_word(“claim”) and has_word(“prize”)) or (has_word(“claim”) and has_word(“FREE”))

Basically pick any combination of words that add up to > 3.099245 (w0)

Perform a parameter sweep on the number of features to use as selected by frequency, using n = [1, 10, 20, 30, 40, 50]. Produce a plot with the number of features used on the X axis, and the train and validation losses plotted on the y axis. Make sure to label the chart correctly and completely! (in the future you will lose 0.5 points for anychart that isn't properly labeled)



Perform a parameter sweep on the number of features to use as selected by mutual information, using n = [1, 10, 20, 30, 40, 50]. Produce a plot with the number of features used on the X axis, and the train and validation losses plotted on the y axis.



Provide short (1-3 sentence) answers to the following:

- **Which feature selection seems better based on the information you have? Why?**
- **Would it make sense to try n = 100 with mutual information based feature selection? Why?**

The mutual information-based model is better as its features are more indicative of spam vs. most frequent words which occur commonly across spam and non-spam messages. Across all number of features, the MI-based model performed better.

I think it would make sense to expand to n=100 as that would bring more words and their associated weights to better segment against false negatives and false positives while strengthening the confidence in true positives and true negatives.