Dataset chosen:

https://www.kaggle.com/datasets/wanderfj/enron-spam?resource=download

Preprocessing:
1.In the dataset which I have used we have 2 folders,spam and ham emails.
2.I made a dictionary of words for training and preprocessed the emails by removing characters and stop words.
3.After iteration through all emails,I have a dictionary of 50500 words.

Training the Model:
1.I have made two dictionaries to store the probability of words being in spam or ham emails spam_vocabulary and ham_vocabulary.
2.If a particular word may not appear in the set of spam or ham emails, I added a mail with all the words that are in the dictionary meaning added 1 to all the values of the spam word vocabulary if there is a word with zero count.
3.phat is the prior probability of an email to be spam or ham.
phat=Number of spam emails/Number of spam + Number of ham emails=0.29

$$P(\text{spam}/\text{email}) = \prod P_k^{f_k} (1-P_k)^{(1-f_k)} \times P_{\text{spam}}$$

P_{spam} is prior probability of email being spam.

P_k is probability of k^{th} word occurring in spam emails.

$f_k = 1$ if word is present in test email.
$= 0$ else.

$$P(\text{ham}/\text{email}) = \prod P_j^{f_j} (1-P_j)^{(1-f_j)} \times P_{\text{nam}}$$

P_{nam} is prior probability of email being ham.

P_j is probability of k^{th} word occurring in ham emails.

$f_k = 1$ if word is present in test email
$= 0$ else.

Prediction for test email:

1.For a given test email , I preprocessed and got the list of words present in that email and created a feature vector. If word is present then feature[i]=1 else 0.
2.Calculated the probability by using above formulas.
3.If ham_probability> spam_probability return ham else return spam.
3.Accuracy = number of spam classified by algorithms/number of emails.

Accuracy on spam emails: 87.46666666666667

Accuracy on ham emails: 82.87037037037037