

# CS471: Introduction to Artificial Intelligence

## Assignment 3: Naive Bayes

In this assignment, you will implement the Naive Bayes classification method using Python.

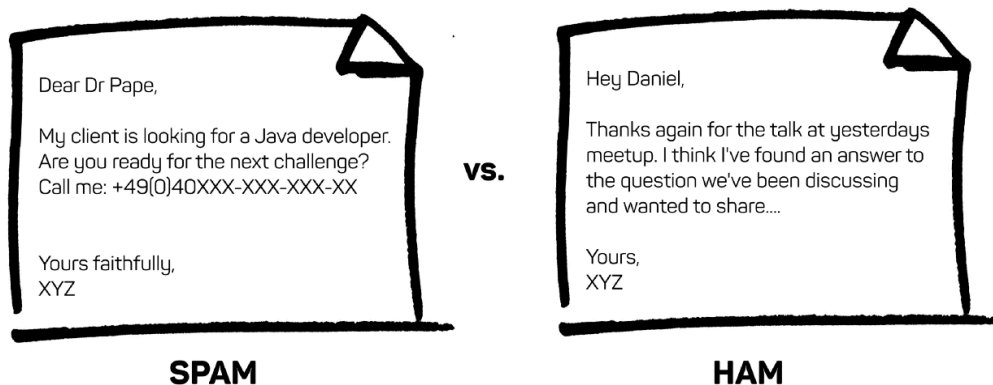
You can either do this project individually or in a group of two. Use Google Colab for your code, plots, and comments. When you finish editing, re-run all the cells to make sure they work.

The Naive Bayes code must be your own. You are not allowed to use any machine learning libraries such as scikit-learn.

---

---

**Goal is to classify a given message either a “spam” or a “ham”, using a Naive Bayes classifier.**



For this assignment, you will be working with a Spam Collection dataset, consisting of text messages that have been collected for Spam research.

Dataset can be downloaded from here:

<https://drive.google.com/file/d/1MJzF39zIYnTkH4SfvW0TcTVXTrTAeDP-/view?usp=sharing>

The csv file contains one message per line with a total of 30 messages tagged either being ham (legitimate) or spam. Each line is composed of two columns: column 1 contains the label (ham or spam) and column 2 contains raw text.

**Consider the first 20 samples as your training set and the rest 10 samples for your testing.**

Tasks:

1. Load the dataset and split into training and testing sets (first 20 into training and the rest into testing) (1 point)
2. Compute the prior probabilities:  $P(\text{spam})$  and  $P(\text{ham})$  (2 points)
3. Compute the conditional probabilities  $P(\text{sentence}/\text{spam})$  (2 points)
4. Compute the posterior probabilities (probability of a sentence belonging to a spam or ham) (2 points)

$$P(\text{spam}/\text{sentence}) \propto P(\text{spam}) * P(\text{sentence}/\text{spam})$$

$$\text{Posterior} \propto \text{prior} * \text{conditional}$$

$$P(\text{ham}/\text{sentence}) \propto P(\text{ham}) * P(\text{sentence}/\text{ham})$$

5. For each sentence in the test set: (2 points)  
Display the sentence  
Print the posterior probability of a sentence belonging to spam or ham  
Display the class (spam or ham)
6. Report the test set accuracy (1 point)  
Accuracy = no. of sentences correctly predicted by model / total sentences

Google Colab Link:

<https://colab.research.google.com/drive/1MxgQpDK7R3alyBEsNFM5JBT5mXNDPb3l?usp=sharing>

GitHub Link:

<https://github.com/jacobalmon/CS-471/tree/main/Homework/Homework%203>

### Task 1:

In this task, we need to extract the raw data into a training and testing dataset. To do this we ignore the entry since it's the header. After that we slice the first 20 entries of the raw data and store it inside of a list for the training data, and the remaining are for the testing data. Here is the following code to perform this:

```
import csv

def split_data(file):
    # Loading the Dataset.
    with open(file, 'r', encoding='latin-1') as file:
        reader = csv.reader(file)
        next(reader) # Skip Header.
        dataset = list(reader) # Convert Dataset to List.

    # Splitting Data
    train_data = dataset[:20] # First 20 Entries.
    test_data = dataset[20:] # Last 10 Entries.

    return train_data, test_data
```

For the implementation, I chose to use the csv module to extract the data.

### Task 2:

In this task, we need to compute the prior probabilities for the sentence being spam or ham. This only applied to the training set, and these probabilities will be used to test the data with the testing dataset. To perform this, we can just count the number of instances where a sentence is spam from the label and divide it by the number of entries in the training data. To find the prior probabilities of ham, we could just do the same thing, but we could just take the compliment from the spam prior probability. So  $P(\text{Spam})$  is 0.45, and  $P(\text{Ham}) = 1 - 0.45$  which is 0.55. Here is the following code segment to achieve this behavior:

```
def prior_probability(train_data):
    # Calculates Sample Space.
    sample_space = len(train_data)
    count_spam = 0

    # Counting the Messages that are Spam.
    for msg in train_data:
        if msg[0] == 'spam':
            count_spam += 1

    # Calculates Probability of Spam.
    prob_spam = count_spam / sample_space
    # Take the Complement to Calculate Probability of Ham.
    prob_ham = 1 - prob_spam

    return prob_spam, prob_ham
```

After we can just return the prior probabilities for later use. Here are the outputs and tests for the following code:

```
pp_spam, pp_ham = prior_probability(train_data)
print(f'Prior Probability of Spam: {pp_spam}\n')
print(f'Prior Probability of Ham: {pp_ham}\n')
```

```
Probability of Spam: 0.45
```

```
Prior Probability of Ham: 0.55
```

### Task 3:

In this task, we need to compute the conditional probabilities of a sentence given its spam or ham. And before we calculate that, we need the frequency of each unique word in the training data on whether it's spam or ham. To perform this, we create two default dictionaries of integers to count the frequency of each word on whether its spam or ham. One dictionary is for counting spam and the other is for counting ham. We also need to keep track of the number of spam words and ham words for the conditional probabilities as well. Here is the code representing that:

```
def probability_words(train_data):
    # Creating Dicts for keeping track of spam and ham frequencies.
    spam_word_count = defaultdict(int)
    ham_word_count = defaultdict(int)
    total_spam_words, total_ham_words = 0, 0
```

After this, we can go through each sentence in the training data and create a list of words for each sentence. Now we check if the sentences are spam or ham by their label and if they are we add them to the dictionary and increase their frequencies. The code segment represents that:

```
# Calculating Frequencies of Words.
for sentence in train_data:
    label, text = sentence
    words = text.split()

    if label == 'spam': # Update words in dict if spam.
        for word in words:
            spam_word_count[word] += 1
            total_spam_words += 1

    elif label == 'ham': # Update words in dict if ham.
        for word in words:
            ham_word_count[word] += 1
            total_ham_words += 1

    return spam_word_count, ham_word_count, total_spam_words, total_ham_words
```

After we have done this process, we return the two dictionaries we created and the total amount of spam and ham words including repeats. Now we compute the conditional probabilities that we have the frequency of each word. We now are going to use the prior probabilities, the frequencies of the words being spam and ham, total spam and ham words, and smoothing of 1. Smoothing is used in the case that a probability is 0 since if one probability is 0, then the rest is. Similar to what we did before we break the sentence into a list of those words. We then set the probability to the prior probability of each like so:

```
def cond_probability(sentence, pw_spam, pw_ham, pp_spam, pp_ham, num_words,
                    total_spam_words, total_ham_words, smoothing=1):
    # Break Sentence into Words.
    words = sentence.split()

    # Temporary Probability for the Sentence.
    p_spam = pp_spam
    p_ham = pp_ham
```

Now we can go through each word in the sentence and multiply to our total conditional probability by adding the frequency of the word by the smoothing and dividing it by total unique

words in the dataset and the total words under that constraint of being spam or ham. The code segment shows the following mentioned:

```
for word in words: # Calculating the Probability by Multiplying by
Prior Probabilities and Word Probabilities.
    p_spam *= (pw_spam.get(word, 0) + smoothing) / (total_spam_words +
num_words)
    p_ham *= (pw_ham.get(word, 0) + smoothing) / (total_ham_words +
num_words)

return p_spam, p_ham
```

Here is how the following task was tested for each sentence in the training data:

```
pw_spam, pw_ham, total_spam_words, total_ham_words =
probability_words(train_data)
num_words = find_num_words(train_data)
for i in range(len(train_data)):
    pg_spam, pg_ham = cond_probability(train_data[i][1], pw_spam,
pw_ham, pp_spam, pp_ham, num_words, total_spam_words, total_ham_words)
    print(f'Sentence: {train_data[i][1]}')
    print(f'Conditional Probability of Sentence Given it's spam:
{pg_spam}\nConditional Probability of Sentence Given it's ham:
{pg_ham}\n')
```

Sentence: WINNER As a valued network customer you have been selected to  
receive a prize reward To claim call

Conditional Probability of Sentence Given it's spam:

1.5186868817999428e-38

Conditional Probability of Sentence Given it's ham: 1.9864617729918437e-41

Sentence: If you had your mobile for more than an year you are entitled to  
Update to the latest colour mobiles with camera for Free Call The Mobile

Conditional Probability of Sentence Given it's spam: 9.412723215017652e-57

Conditional Probability of Sentence Given it's ham: 1.0830558046385033e-62

Sentence: I am gonna be home soon

Conditional Probability of Sentence Given it's spam: 5.825980238523606e-16

Conditional Probability of Sentence Given it's ham: 7.329851462608767e-13

Sentence: SIX chances to win CASH From 100 to 20000 dollars text or call

Conditional Probability of Sentence Given it's spam: 5.520174591825462e-28

Conditional Probability of Sentence Given it's ham: 2.0621586450560087e-31

Sentence: URGENT You have won a 1 week FREE membership in our million  
dollar Prize Jackpot Text the word CLAIM

Conditional Probability of Sentence Given it's spam: 7.146761796705615e-42

Conditional Probability of Sentence Given it's ham: 4.2941240228963337e-45

Sentence: You have been wonderful and a blessing at all times

Conditional Probability of Sentence Given it's spam:

2.6158003250037136e-24

Conditional Probability of Sentence Given it's ham: 1.0081239465962075e-21

Sentence: XXXMobileMovieClub To use your credit click the WAP link in the  
next txt message or click here

Conditional Probability of Sentence Given it's spam:

1.2392484955487535e-36

Conditional Probability of Sentence Given it's ham: 1.8908173172551997e-40

Sentence: okay I am watching here

Conditional Probability of Sentence Given it's spam: 1.980833281098026e-13

Conditional Probability of Sentence Given it's ham: 9.418859129452265e-11

Sentence: England vs Macedonia dont miss the team news Text ur national  
team

Conditional Probability of Sentence Given it's spam:

1.1730371007629113e-26

Conditional Probability of Sentence Given it's ham: 1.3249369294484854e-29

Sentence: Is that seriously how you spell his name

Conditional Probability of Sentence Given it's spam:

1.2599438232101224e-20

Conditional Probability of Sentence Given it's ham: 3.329278927436645e-17

Sentence: did you finish your lunch already

Conditional Probability of Sentence Given it's spam:  $7.282475298154506e-15$

Conditional Probability of Sentence Given it's ham:  $5.497388596956575e-13$

Sentence: Alright no way I can meet up with you sooner

Conditional Probability of Sentence Given it's spam:

$2.1798336041697616e-25$

Conditional Probability of Sentence Given it's ham:  $4.032495786384829e-21$

Sentence: Just forced myself to eat a slice I am really not hungry

Conditional Probability of Sentence Given it's spam:  $8.485511434916891e-30$

Conditional Probability of Sentence Given it's ham:  $2.442123748359448e-25$

Sentence: Did you catch the train

Conditional Probability of Sentence Given it's spam:  $2.971249921647039e-12$

Conditional Probability of Sentence Given it's ham:  $4.7094295647261327e-11$

Sentence: tell me anything about you.

Conditional Probability of Sentence Given it's spam:  $9.90416640549013e-14$

Conditional Probability of Sentence Given it's ham:  $1.5698098549087107e-11$

Sentence: Thanks for your subscription to Ringtone UK your mobile will be charged Please confirm by replying YES or NO

Conditional Probability of Sentence Given it's spam:

$1.1117185017097623e-40$

Conditional Probability of Sentence Given it's ham:  $2.862749348597556e-45$

Sentence: Hello How are you and how did saturday go

Conditional Probability of Sentence Given it's spam:

$1.4822868508354382e-22$

Conditional Probability of Sentence Given it's ham:  $5.829476721192569e-19$

Sentence: Rodger Burns MSG We tried to call you reply to our sms for a free samsung mobile and free camcorder Please call now for delivery tomorrow

Conditional Probability of Sentence Given it's spam:  $9.720989900309483e-54$

Conditional Probability of Sentence Given it's ham:  $1.0437950317203573e-60$



```
Sentence: Congrats one year special cinema pass for two is yours call now  
Dont miss out
```

```
Conditional Probability of Sentence Given it's spam: 5.571110458878063e-33
```

```
Conditional Probability of Sentence Given it's ham: 7.805412061711792e-37
```

```
Sentence: Sorry I will call later in meeting
```

```
Conditional Probability of Sentence Given it's spam: 3.084342479218379e-17
```

```
Conditional Probability of Sentence Given it's ham: 2.8520822811707264e-15
```

#### Task 4:

In this task, we need to compute the posterior probabilities which the probability of a sentence belonging to spam or ham. Basically, we are finding the probability of a sentence given it's spam and the probability of a sentence given it's ham. We have already implemented the conditional probabilities, but how we test it is a little different. Firstly we would find these conditional probabilities as before, and we are essentially multiplying the prior probabilities and the conditional probabilities to each other to give us our posterior probabilities. Here is the following code segment to test and find the posterior probabilities:

```
responses_train = []  
num_sentence_train = len(train_data)  
for i in range(len(train_data)):  
    pg_spam, pg_ham = cond_probability(train_data[i][1], pw_spam, pw_ham,  
    pp_spam, pp_ham, num_words, total_spam_words, total_ham_words)  
    pgs_sentence = pp_spam * pg_spam  
    posterior_spam = pp_spam * pg_spam  
    posterior_ham = pp_ham * pg_ham  
    pgh_sentence = pp_ham * pg_ham  
  
    if pgs_sentence > pgh_sentence:  
        classType = 'spam'  
    else:  
        classType = 'ham'  
    responses_train.append(classType)  
  
    print(f'Sentence: {train_data[i][1]}')  
    print(f'Posterior Probability of Spam Given a Sentence:  
{pgs_sentence}')  
    print(f'Posterior Probability of Spam: {posterior_spam}')  
    print(f'Posterior Probability of Ham: {posterior_ham}')  
  
print(f'Posterior Probability of Ham
```

Here are the outputs for the following test:

Sentence: WINNER As a valued network customer you have been selected to receive a prize reward To claim call

Posterior Probability of Spam Given a Sentence:  $6.834090968099743e-39$

Posterior Probability of Spam:  $6.834090968099743e-39$

Posterior Probability of Ham:  $1.0925539751455141e-41$

Posterior Probability of Ham Given a Sentence:  $1.0925539751455141e-41$

Sentence: If you had your mobile for more than an year you are entitled to Update to the latest colour mobiles with camera for Free Call The Mobile

Posterior Probability of Spam Given a Sentence:  $4.2357254467579434e-57$

Posterior Probability of Spam:  $4.2357254467579434e-57$

Posterior Probability of Ham:  $5.956806925511769e-63$

Posterior Probability of Ham Given a Sentence:  $5.956806925511769e-63$

Sentence: I am gonna be home soon

Posterior Probability of Spam Given a Sentence:  $2.621691107335623e-16$

Posterior Probability of Spam:  $2.621691107335623e-16$

Posterior Probability of Ham:  $4.0314183044348223e-13$

Posterior Probability of Ham Given a Sentence:  $4.0314183044348223e-13$

Sentence: SIX chances to win CASH From 100 to 20000 dollars text or call

Posterior Probability of Spam Given a Sentence:  $2.484078566321458e-28$

Posterior Probability of Spam:  $2.484078566321458e-28$

Posterior Probability of Ham:  $1.1341872547808048e-31$

Posterior Probability of Ham Given a Sentence:  $1.1341872547808048e-31$

Sentence: URGENT You have won a 1 week FREE membership in our million dollar Prize Jackpot Text the word CLAIM

Posterior Probability of Spam Given a Sentence:  $3.216042808517527e-42$

Posterior Probability of Spam:  $3.216042808517527e-42$

Posterior Probability of Ham:  $2.3617682125929838e-45$

Posterior Probability of Ham Given a Sentence:  $2.3617682125929838e-45$

Sentence: You have been wonderful and a blessing at all times

Posterior Probability of Spam Given a Sentence:  $1.1771101462516711e-24$

Posterior Probability of Spam:  $1.1771101462516711e-24$

Posterior Probability of Ham:  $5.544681706279142e-22$

Posterior Probability of Ham Given a Sentence:  $5.544681706279142e-22$

Sentence: XXXMobileMovieClub To use your credit click the WAP link in the next txt message or click here

Posterior Probability of Spam Given a Sentence:  $5.576618229969391e-37$

Posterior Probability of Spam:  $5.576618229969391e-37$

Posterior Probability of Ham:  $1.03994952449036e-40$

Posterior Probability of Ham Given a Sentence:  $1.03994952449036e-40$

Sentence: okay I am watching here

Posterior Probability of Spam Given a Sentence:  $8.913749764941118e-14$

Posterior Probability of Spam:  $8.913749764941118e-14$

Posterior Probability of Ham:  $5.1803725211987466e-11$

Posterior Probability of Ham Given a Sentence:  $5.1803725211987466e-11$

Sentence: England vs Macedonia dont miss the team news Text ur national team

Posterior Probability of Spam Given a Sentence:  $5.278666953433101e-27$

Posterior Probability of Spam:  $5.278666953433101e-27$

Posterior Probability of Ham:  $7.28715311196667e-30$

Posterior Probability of Ham Given a Sentence:  $7.28715311196667e-30$

Sentence: Is that seriously how you spell his name

Posterior Probability of Spam Given a Sentence:  $5.669747204445551e-21$

Posterior Probability of Spam:  $5.669747204445551e-21$

Posterior Probability of Ham:  $1.831103410090155e-17$

Posterior Probability of Ham Given a Sentence:  $1.831103410090155e-17$

Sentence: did you finish your lunch already

Posterior Probability of Spam Given a Sentence:  $3.277113884169528e-15$

Posterior Probability of Spam:  $3.277113884169528e-15$

Posterior Probability of Ham:  $3.023563728326116e-13$

Posterior Probability of Ham Given a Sentence:  $3.023563728326116e-13$

Sentence: Alright no way I can meet up with you sooner

Posterior Probability of Spam Given a Sentence:  $9.809251218763927e-26$

Posterior Probability of Spam:  $9.809251218763927e-26$

Posterior Probability of Ham:  $2.2178726825116563e-21$

Posterior Probability of Ham Given a Sentence:  $2.2178726825116563e-21$

Sentence: Just forced myself to eat a slice I am really not hungry

Posterior Probability of Spam Given a Sentence:  $3.8184801457126015e-30$

Posterior Probability of Spam:  $3.8184801457126015e-30$

Posterior Probability of Ham:  $1.3431680615976965e-25$

Posterior Probability of Ham Given a Sentence:  $1.3431680615976965e-25$

Sentence: Did you catch the train

Posterior Probability of Spam Given a Sentence:  $1.3370624647411675e-12$

Posterior Probability of Spam:  $1.3370624647411675e-12$

Posterior Probability of Ham:  $2.5901862605993733e-11$

Posterior Probability of Ham Given a Sentence:  $2.5901862605993733e-11$

Sentence: tell me anything about you.

Posterior Probability of Spam Given a Sentence:  $4.456874882470559e-14$

Posterior Probability of Spam:  $4.456874882470559e-14$

Posterior Probability of Ham:  $8.63395420199791e-12$

Posterior Probability of Ham Given a Sentence:  $8.63395420199791e-12$

Sentence: Thanks for your subscription to Ringtone UK your mobile will be charged Please confirm by replying YES or NO

Posterior Probability of Spam Given a Sentence:  $5.00273325769393e-41$

Posterior Probability of Spam:  $5.00273325769393e-41$

Posterior Probability of Ham:  $1.574512141728656e-45$

Posterior Probability of Ham Given a Sentence:  $1.574512141728656e-45$

Sentence: Hello How are you and how did saturday go

Posterior Probability of Spam Given a Sentence:  $6.670290828759472e-23$

Posterior Probability of Spam:  $6.670290828759472e-23$

Posterior Probability of Ham:  $3.2062121966559133e-19$

Posterior Probability of Ham Given a Sentence:  $3.2062121966559133e-19$

Sentence: Rodger Burns MSG We tried to call you reply to our sms for a free samsung mobile and free camcorder Please call now for delivery tomorrow

Posterior Probability of Spam Given a Sentence: 4.3744454551392674e-54

Posterior Probability of Spam: 4.3744454551392674e-54

Posterior Probability of Ham: 5.7408726744619655e-61

Posterior Probability of Ham Given a Sentence: 5.7408726744619655e-61

Sentence: Congrats one year special cinema pass for two is yours call now Dont miss out

Posterior Probability of Spam Given a Sentence: 2.5069997064951283e-33

Posterior Probability of Spam: 2.5069997064951283e-33

Posterior Probability of Ham: 4.292976633941486e-37

Posterior Probability of Ham Given a Sentence: 4.292976633941486e-37

Sentence: Sorry I will call later in meeting

Posterior Probability of Spam Given a Sentence: 1.3879541156482705e-17

Posterior Probability of Spam: 1.3879541156482705e-17

Posterior Probability of Ham: 1.5686452546438995e-15

Posterior Probability of Ham Given a Sentence: 1.5686452546438995e-15

#### Task 5:

In this task, we test our ML model on whether a sentence is spam or ham. Similar to the previous task, the same process is used, but instead of applying it to the training data, we are applying it to the testing data. Here is how the following code segment to test this task:

```
test_num_words = find_num_words(test_data)
num_sentence_test = len(test_data)
responses_test = []
for i in range(num_sentence_test):
    pgs_sentence, pgh_sentence = cond_probability(test_data[i][1], pw_spam,
    pw_ham, pp_spam, pp_ham, num_words, total_spam_words, total_ham_words)

    if pgs_sentence > pgh_sentence:
        classType = 'spam'
        probab = pgs_sentence
    else:
        classType = 'ham'
```

```

        prob = pgh_sentence
        responses_test.append(classType)

    print(f'Sentence: {test_data[i][1]}')
    print(f'Posterior Probability of Sentence being Spam: {pgs_sentence}')
    print(f'Posterior Probability of Sentence being Ham: {pgh_sentence}')
    print(f'The Sentence is {classType}\n')

```

Here are the outputs for testing the model:

Sentence: Tell where you reached

Posterior Probability of Sentence being Spam: 1.683708288933322e-10

Posterior Probability of Sentence being Ham: 7.564521238341349e-10

The Sentence is ham

Sentence: Your gonna have to pick up a burger for yourself on your way home

Posterior Probability of Sentence being Spam: 7.707428206455655e-33

Posterior Probability of Sentence being Ham: 3.8515025277310664e-32

The Sentence is ham

Sentence: As a valued customer I am pleased to advise you that for your recent review you are awarded a Bonus Prize

Posterior Probability of Sentence being Spam: 1.956571870180832e-47

Posterior Probability of Sentence being Ham: 1.6851685063130853e-46

The Sentence is ham

Sentence: Urgent you are awarded a complimentary trip to EuroDisinc To claim text immediately

Posterior Probability of Sentence being Spam: 2.994886388794197e-30

Posterior Probability of Sentence being Ham: 1.8559427805504077e-30

The Sentence is spam

Sentence: Finished class where are you

Posterior Probability of Sentence being Spam: 9.90416640549013e-13

Posterior Probability of Sentence being Ham: 5.886786955907665e-12

The Sentence is ham

Sentence: where are you how did you perform

```
Posterior Probability of Sentence being Spam: 4.2838089989144154e-17
Posterior Probability of Sentence being Ham: 4.8128888494755994e-15
The Sentence is ham
```

```
Sentence: you can call me now
Posterior Probability of Sentence being Spam: 8.913749764941117e-12
Posterior Probability of Sentence being Ham: 2.3547147823630663e-11
The Sentence is ham
```

```
Sentence: I am waiting Call me once you are free
Posterior Probability of Sentence being Spam: 4.446860552506315e-22
Posterior Probability of Sentence being Ham: 6.477196356880635e-20
The Sentence is ham
```

```
Sentence: I am on the way to homei
Posterior Probability of Sentence being Spam: 4.626513718827569e-17
Posterior Probability of Sentence being Ham: 1.4260411405853632e-15
The Sentence is ham
```

```
Sentence: Please call our customer service representative between 10am-9pm
as you have WON a guaranteed cash prize
Posterior Probability of Sentence being Spam: 2.2859401502092896e-37
Posterior Probability of Sentence being Ham: 1.0933651137028192e-37
The Sentence is spam
```

#### Task 6:

In this task, we are evaluating the ML model's performance, in the previous two tasks, we keep track of the responses to get a ratio of how accurate the model is. Here is how the accuracy is calculated which is just dividing the number of correct responses by the total number of sentences in either training or testing data. Here is the code segment to perform this:

```
def test_prediction(responses, data, num_sentence):
    correct = 0
    # Finding the Accuracy Given Either training data or testing data.
    for i in range(num_sentence):
        # Update correctness if the data and the response matches.
        if responses[i] == data[i][0]:
            correct += 1
```

```
return correct / num_sentence
```

Here is how the following code is being tested:

```
accuracy_train = test_prediction(responses_train, train_data,  
num_sentence_train)  
accuracy_test = test_prediction(responses_test, test_data,  
num_sentence_test)  
print(f'Training Accuracy: {accuracy_train}')print(f'Testing Accuracy: {accuracy_test}')
```

Here are the outputs of the accuracy:

```
Training Accuracy: 1.0
```

```
Testing Accuracy: 0.9
```

As we can see our ML model is 90% accurate in testing which is very good to detect spam like sentence, especially from the email or text standpoint.