Name : Rutuj Kuldeep Khare
Roll no. : CS22M074

ASSIGNMENT 3

1. Dataset

   To implement a spam classifier, we are going to use the SMSSpamCollection dataset consisting of 5,574 messages, tagged according to being ham or spam.

2. Preprocessing
   - Before preprocessing the dataset, we split the dataset into a training dataset and test dataset in the ratio of 80:20 after randomizing the dataset to ensure that spam and ham messages are properly spread throughout the dataset.

```
ds = dataset.sample(frac=1, random_state=1)
idx = round(len(ds)*0.8)
train_data = ds[:idx].reset_index(drop=True)
test_data = ds[idx:].reset_index(drop=True)
```

   - Then we will begin the data cleaning process by removing the punctuation along with making all words in lowercase.

```
train_data['Email'] = train_data['Email'].str.replace('\W', ' ')
train_data['Email'] = train_data['Email'].str.lower()
```

   - We will transform our dataset into the following format
     - The first column will represent the index of the message.
     - The second column will represent the label of the message ie. whether the message is spam or ham.
     - We will replace the SMS column with a series of new columns which will represent the unique words from each message that is present in our dataset.

	Label	secret	prize	claim	now	coming	to	my	party	winner
0	spam	2	2	1	1	0	0	0	0	0
1	ham	1	0	0	0	1	1	1	1	0
2	spam	1	1	1	1	0	0	0	0	1

3. Naive Bayes
   - To build the spam classifier, we will use the Naive Bayes algorithm which is the most popular algorithm.
   - If the probability of a message being spam, given the series of words is greater than the probability of a message being spam, given the series of words, then we conclude that the given message is a spam message. Otherwise, it is a ham message.

- To calculate these probabilities, we have to find the following parameters:
    - Probability of a specific word given that it is present in the spam message.
    - Probability of a specific word given that it is present in the ham message.

$$P(\text{Spam}|w_1, w_2, ..., w_n) \propto P(\text{Spam}) \cdot \prod_{i=1}^{n} P(w_i|\text{Spam})$$

$$P(\text{Ham}|w_1, w_2, ..., w_n) \propto P(\text{Ham}) \cdot \prod_{i=1}^{n} P(w_i|\text{Ham})$$

- Following is the general algorithm:
    - Take input message as (w_1, w_2, … w_i)
    - Calculate P(Spam | w_1, w_2, … w_i) and P(Ham | w_1, w_2, … w_i).
    - Compare values of P(Spam|w_1, w_2, … w_i) and P(Ham|w_1, w_2,...w_i)
    - If P(Spam|w_1, w_2, … w_i) > P(Ham|w_1, w_2,...w_i), then the message is classified as a spam message, otherwise, the message is classified as a ham message.

4. Accuracy

    After applying the Naive Bayes classifier to our test dataset, we got an accuracy of 98.743 which is pretty good.