The objective of the agent is to distinguish spam from normal emails. Normal email is labelled as “1” and spam as “0”.

I have applied the Naive Bayes algorithm to implement this classifier. This is a "probabilistic classifiers" based on applying Bayes' theorem with independence assumptions between the features. In the case, independence means that the words themselves (spam/ham) are independent of each other.

The agent first defines a numpy array filled with zeros. After, the agent calculates the logarithm of the probability of the class occurring. To make the training easier, training data is separated into spam and ham. Using a for loop the agent calculates theta, it iterates over the features (words) to calculate the probability for each feature to be a spam or ham. Each theta value is assigned to each word and it is stored in numpy array (theta[][]). In each loop the probability is reset for each message either it is a ham or spam.

theta value for this word given the class multiplied by the occurrence of the word (1 if present, 0 if not). to the possibility that the communication falls into the category of (ham/spam). To compare the two later, I created distinct variables for the probability of ham and spam. There is a condition at the conclusion of the same for-loop to determine whether the message is spam.

This classifier method achieved accuracy at around 90%.  accuracy will be around 90% going forward because of this. Since most emails will be correctly categorized, this accuracy is acceptable.

Adding more data to the training set to improve the classification model's accuracy would considerably improve the accuracy. Making the classifier more trustworthy since there will be more evidence to back up its predictions, providing users greater assurance.