ASSIGNMENT-3 REPORT

ED20B026
KAJA VIKAS

https://www.kaggle.com/datasets/venky73/spam-mails-dataset
Used dataset from this website spam_ham_dataset.csv for training the data

Dataset has row containing label(spam or ham) and email text and label_num(0 or 1)
We will drop the rows with null values

Preprocessing training data:
Check if the input is a string.
Removes all punctuation marks from the text using the translate() method and string.punctuation module.
Converts all text to lowercase using the lower() method.
Removes all stop words such as {a,the,these…} which have no major importance in emails using the stopwords module from the NLTK library.
Join the remaining words back together into a string.

After preprocessing we split the dataset into train and test

Algorithm:naive bayes

spam_words and ham_words are initialized as empty dictionaries that will store the count of each word for each class(spam and ham)
spam_total_words and ham_total_words are initialized to zero and will store the total number of words in each class.

calculating the prior probabilities of each class.
The prior probability is the probability of an email being spam or ham before looking at any features or words in the email. We can calculate the prior probabilities by dividing the number of spam or ham emails in the training set by the total number of emails in the training set.

We iterate over each email in the test data.
Initialize the scores for both the 'spam' and 'ham' classes to 0.
Iterate over each word in the test email.
Calculate the count of the word in both the 'spam' and 'ham' classes using the word count dictionaries created during training.
If the word is not present in the dictionary for a class, its count is assumed to be 0.
Calculate the log probabilities of the word being present in each class using Laplace smoothing.
Probability(word|spam(ham))=no of times the word occurred in spam(ham) emails/no of times the word occurred in all training emails

The laplace smoothed probabilities are calculated by adding 1 to the count of the word in the class and dividing by no of times the word occurred in all training emails+1.

Laplace smoothing adds a small positive value (usually 1) to the count of each word in the vocabulary, so that even if a word is not present in the training set, it will still have a non-zero probability of occurring in a document. This is important because if a word has a probability of 0, then it will always lead to a 0 probability for the entire document, regardless of the other words in the document. This can lead to incorrect classification of documents.

Add the log prior probabilities of each class to the respective scores.

Classify the email as 'spam' if the spam score is higher than the ham score, else as 'ham'.

Here we mainly compare the frequency of test email word occurrences in spam mails and ham mails and based on this we classify mails as spam or ham.

Here I have created my own folder with the testing emails I have and given this as a path to the code which reads test emails and classifies the emails.

In this assignment code asks for a path to give as input to read the emails in the directory and later it prints email name and corresponding classification(1 or 0).here for convenience i had created a dataframe with the emails in directory and classified them as spam or ham using the above algorithm.algorithm gives 92%accuracy