Pattern Recognition And Machine Learning - Assignment 3

Subject - Spam Classifier

Dataset:

The dataset used for the project is the “Spam Ham Dataset” which is publicly available at kaggle on the following link:
https://www.kaggle.com/datasets/venky73/spam-mails-dataset
The dataset is such that it contains a csv file containing majorly 4 columns.

- Column 1 gives a number to each unique email.
- Column 2 is basically a textual label as to which is a spam mail and which is a ham mail(i.e not spam mail)
- Column 3 is containing the entire email
- Column 4 is basically containing the numerical label(i.e 1 if the mail is spam, 0 otherwise)

The dataset contains in total 4993 unique emails.

The distribution of the dataset is as follows:

- 71% Ham i.e not spam mails
- 29% Spam mails

The class labels are as follows

- 1 if the given email is a spam
- 0 if the given email is not a spam

The above data set is used to create a dictionary for all the valid words. This dataset is used to train the model and learn the different probabilities for different words i.e probability of word given mail is spam and the probability of word given the mail is ham.

Feature Extraction:

The features or the dimensions of our dictionary were derived from the above spam ham dataset. We create a dictionary named collection to store all the unique valid words from the email which will serve as a word corpus for any upcoming test mail.
We use the NLTK library to classify whether a word is actually useful i.e whether it is a valid english word or not.
Procedure:

- Read the dataset from the csv file
- For each email split the email into an array which contains all the words contained in it
- For each word so obtained in the array we check if the word is not already present in our dictionary and it is a valid word then add it in the dictionary and assign a unique value to this word

This way we were able to generate approximately 18000 words.

Parameters:

For each word in our dictionary we calculate the probability of a word being present in the mail when the mail is known to be spam and also the probability of the word being present in the mail when the mail is known to be not spam. This if we have x number of words in our dictionary then we need 2*x number of probability values for words in the dictionary.

Additionally, there is one more parameter p which is the probability of a mail being spam or not. Thus in total there are 2*x+1 parameters to estimate.

Estimation of the parameters:

The 2*x+1 parameters are estimated using the maximum likelihood estimation technique.

- p=number of spam mails / (number of spam mails+number of non spam mails)
- p(x/1)=number of spam emails in which word x is present / number of spam emails
- p(x/0)=number of non spam emails in which word x is present / number of non spam emails

Algorithm used:

The following project used the Naive Bayes algorithm which predicts whether an email is a spam or not by calculating values of two probability which are:

- Probability of mail being a spam
- Probability of mail not being a spam

For doing so we consider each word in our dictionary and then check if that word is also present in our test email then we multiply with the probability of it being there given we are calculating for being a spam or not spam case.

Prediction:

The prediction is based on the result of the following log value. If the value is greater than 0 then we predict the spam as spam else not spam.

$$log(\frac{P(pos)}{P(neg)}\prod_{i=1}^{n}\frac{P(w_i|pos)}{P(w_i|neg)}) \Rightarrow \underbrace{log\frac{P(pos)}{P(neg)}}_{\textbf{log prior}} + \underbrace{\sum_{i=1}^{n} log\frac{P(w_i|pos)}{P(w_i|neg)}}_{\textbf{log likelihood}}$$

- Log value greater than 0 then spam
- Log value less than 0 then non spam