Name : Aashay Shah **Roll No : CS22M004**

PRML Assignment 3

Aim : Build a Spam Classifier for emails using labeled data.

Dataset

The dataset used for this task is attached with the zip file, which contains approximately 6000 emails in total. But all of them are not used for training. For testing purposes, I have used others' emails and put them into the folder named 'Test'.

Models that I considered for this task

Naive Bayes : This algorithm is very simple since it is using the assumption that all the features are independent, but practically this might not be the case. This algorithm gives the same importance to all the words, but some words are practically more important to detect whether the mail is spam or not spam.

SVM with Polynomial and Radial basis kernel

I have implemented the Support vector Machine algorithm with polynomial and radial basis kernels. I have trained by Polynomial kernel with degrees 2 and 3 and radial basis kernel with regularized parameter value 3.0

Feature Extractor

I have used CountVectorizer and Term frequency-inverse document frequency. We have to count the frequency of each word appearing in the dataset. But to use alone this library might not give a better result because it is not giving importance to the words according to their meaning. e.g money, lottery words are highly related to the scam and so the mail might be spam.

Procedure

- First of all, from the given dataset we have to remove some of the columns that are not required for the task. The target column is categorical and if it is not in the form of 1/0, we have to map the spam label to 1 and the non-spam label to 0. After doing this task, we have to perform some preprocessing work.

- Preprocessing task includes lowering all the sentences of the email, removing the stopwords, a punctuation marks from the email, and removing the extra spaces. I have used a regular expression for replacing some of the things like, If some website link is given then, replace that part with the word 'webaddress', replace the 10 digit

phone number with the word ‘phone number’, replace any other number by the word ‘number’, replace any email address by the word ‘email address.

- After this task, I performed the train-test split with 75% for training data and 25% for validation data. Now we will give this training data to our feature extractor count vectorizer and tfidf. Now we have data ready for performing the actual training task. I have performed training on naïve Bayes, and SVM algorithms but the code is only shown for SVM only. I have hyperparameterized this algorithm by parameter regularization in radial basis kernel and degrees 2 and 3 for the polynomial kernel.

- Here, the Radial basis kernel with regularization parameters 3.0 gives the best accuracy out of all the other algorithms. After performing the training, I saved the best model with the feature extractor.

Prediction

After performing the training and validation, Now Testing is performed on the test data store inside the folder named ‘Test’. Which contains the new emails to be predicted. I have made a separate file for performing the prediction on test data named ‘Test.py’. which contains two functions, one is for performing the preprocessing task the same way as we did at the time for training. And the other function will load the models saved at the time of training.