Problem Report

Firt of all, the data set I have used for training and testing in 70-30 split is spam_or_not_spam.csv. This data set was downloaded from kaggle. The folder named as test contains 150 emails is in my current directory *C:\Users\Pritam1\test*. Each email in the folder is named as 'email1.txt', 'email2.txt' and so on. The drive link of the files are added in red color, can be accessed through click.

The first thing for the code is to import necessary libraries. I have imported os for handling the folder with .txt files in current directory, imported numpy for numerical operations(for creat, manipulate arrays), imported pandas for handling csv file. From sklearn.svm I have imported SVC(support vector classifier), and necessary library for train-test split of the dataset.

Then I have defined some keywords that are likely to be present in a spam email in practice. Then I have defined a function to clean the data. i.e. all the characters is converted to lowercase, kept the numbers, and all others are seplaced by a space. The data file contains a column having email contents, and another column contains the labels of the emails as 0(non-spam), 1(spam). Each row of the data file is separated into (email, label) format. there were some miisng labels in the dataset, those have been excluded. Then a feature matrix was created, where each row of the matrix is a feature vector.

In this classifier problem, I am using Naive Bayes' Classifier, and SVM classifier. The Naive Bayes' Classifier was implemented from scratch, and SVM was implemented by inbuilt libraries as mentioned in the question.
For Naive Bayes's I have defined a function *calculate_prior* to find prior of the spam emails. The I have added two pseudo emails for Laplace Smoothing through the function *calculate_word_probs*. Then log probability for spam non spam was calculated by calculate_log_probs function. Then the predicted values of the test email is defined through the function *predict*. Finally the Naive Bayes was trained. Then features were extracted for training and testing, and finally the accuracy is calculated.

Then the SVM classifier was applied from the inbuilt libraries. Like before, the SVM classifier was trained, and tested and calculated it's accuracy in prediction.

The the folder 'test' in my current directory was read by the function *read_emails_from_folder*. The emails in the folder was sorted by a lambda function so that the code reads the emails serially('email1.txt', 'email2.txt' and so on). The the function *classify_emails_in_folder* classifies emails from the folder using Naive Bayes, SVM and a csv file *email_predictions.csv* stores the predicted labels of the emails. Then finally the code prints first 15 predicted labels from the folder.

Results

The accuracy in the Naive Bayes's classifier is observed as 88.44%, and in SVM it is 93.11%. The first 15 predictions from the folder is also printed.