Name: Arvind Kumar Gupta
Roll No.: 18AT91R02
Assignment No.: 5

Methodology (Details of SVM package Used): In the email classification using Support Vector Machine, I have used 'scikit-learn' library. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.
Following functions have been used in the email classification:

train_test_split (*arrays, **options): Split arrays or matrices into random train and test subsets

SVC: SVC (C=abc, kernel= 'xyz', degree= int). I have taken the different value of C which is given in the experimental result table. For kernel, 'linear', 'poly', and 'rbf' have been used. For Quadratic kernel, 'poly' kernel has been used with degree = 2.

fit (attributrLevel_train, classLevel_train): Fit the SVM model according to the given training data

predict(attributrLevel_test): Perform classification on samples in attributrLevel_test.

accuracy_score (classLevel_test, y_pred): Returns the mean accuracy on the given test data and labels.

The python program is given as follow:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
dataSet = pd.read_csv('spambase.csv', header=None)
dataSet = dataSet.values
dataSize = dataSet.shape
attributrLevel = dataSet[:,0:57]
classLevel = dataSet[:,57]
attributrLevel_train, attributrLevel_test, classLevel_train, classLevel_test = train_test_split(attributrLevel, classLevel, test_size = 0.30)
genC = [0.1, 0.5, 0.9, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25, 30, 40, 50, 60, 70, 80, 90, 100]
for kernelD in ('rbf', 'linear', 'poly'):
    print(kernelD)
    for gC in genC:
        svclassifier = SVC(C=gC, kernel=kernelD, degree=2)
        svclassifier.fit(attributrLevel_train, classLevel_train)
        y_pred = svclassifier.predict(attributrLevel_test)
        print(accuracy_score(classLevel_test, y_pred))
```

Experimental Result:

I have taken randomly 70% of the data set as training data and the remaining data set as the test data. By varying the value of C, test set classification accuracy has been given in Table 2 for all the three said kernels, i.e. 'linear', 'quadratic', 'rbf'.
The best C value and the best test set accuracy that is have found out by trial of different values of C are given in Table 1.

Table 1 the best value of C

Kernel	Linear	Quadratic	RBF
C	3	2	4
Accuracy	0.930485155684	0.93220258424586251	0.853729181752

Table 2 classification accuracy corresponding to C value

C	Linear	Quadratic	RBF
0.1	0.928312816799	0.92976104272266469	0.742939898624
0.5	0.929036929761	0.93018458968685849	0.788559015206
0.9	0.929761042723	0.9304851556842868	0.823316437364
1	0.929036929761	0.93120926864590881	0.832729905865
2	0.927588703838	0.93220258424586251	0.845763939175
3	0.930485155684	0.93110458256156456	0.852280955829
4	0.929761042723	0.93110247584778541	0.853729181752
5	0.929036929761	0.92110912878617561	0.853005068791
6	0.922519913106	0.92458504287557274	0.851556842867
7	0.92396813903	0.92438247582586557	0.848660391021
8	0.922519913106	0.92434585288778888	0.847212165098
9	0.91962346126	0.92387877587788755	0.847212165098