Support vector machine classifier

Aim: spam email classification using Support Vector machine and report the classification accuracy for various SVM parameters and kernel functions.

Dataset: An email is represented by various features like frequency of occurrences of certain keywords, length of capitalized words etc. A data set containing about 4601 instances are available in this link (data folder):Spambase(data folder)

Dataset description:

Number of Instances: 4601 (1813 Spam = 39.4%)
Number of Attributes: 58 (57 continuous, 1 nominal class label)

- The last column of 'spambase.data' denotes whether the email was considered spam(1) or not (0).
- 48 continuous real [0,100] attributes of type word_freq_WORD = percentage of words in the email that match WORD.
- 6 continuous real [0,100] attributes of type char_freq_CHAR = percentage of characters in the email that match CHAR.
- 1 continuous real [1,...] attribute of type capital_run_length_average = average length of uninterrupted sequences of capital letters.
- 1 continuous integer [1,...] attribute of type capital_run_length_longest = length of longest uninterrupted sequence of capital letters.
- 1 continuous integer [1,...] attribute of type capital_run_length_total = sum of length of uninterrupted sequences of capital letters
- 1 nominal {0,1} class attribute of type spam = denotes whether the email was considered spam (1) or not (0).

Package used:
-> we have used scikit-learn package to implement svm classifier.

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language.[3] 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.

Splitting Data:
We Split the dataset by using the function train_test_split(). you need to pass 3 parameters features, target, and test_set size. Additionally, you can use random_state to select records randomly.

Generating model:

Let's build a support vector machine model. First,we import the SVM module and create a support vector classifier object by passing the argument kernel and generalization constant in the SVC() function.
Then, fit our model on the train set using fit() and perform prediction on the test set using predict().

Evaluating the model:

We are comparing on the basis of accuracy of the model.Accuracy can be computed by comparing actual test set values and predicted values.

Importance of generalization constant:

Large Value of parameter C => small margin
Small Value of parameter C => Large margin

Comparison between different kernel functions:

1. **Linear**

Generalization constant(c)	Accuracy (in decimal)(1 being highest)(range of 0-1)
0.01	0.9102099927588704
0.10	0.9203475742215785
1.00	0.9268645908761767
10.00	0.9268645908761767
100.00	0.9210716871832005
1000.00	0.9210716871832005

We can see that best accuracy comes for c value equal to 1 and 10 and accuracy is 92.68% for maximum.

2. **RBF**

Generalization constant(c)	Accuracy (in decimal)(1 being highest)(range of 0-1)
0.01	0.6705286024619841
0.10	0.6951484431571325
1.00	0.719044170890659

10.00	0.7299058653149891
100.00	0.8124547429398986
1000.00	0.9058653149891384
10000.00	0.9290369297610427
100000.00	0.9326574945691528
1000000.00	0.9326574945691528
10000000.00	0.9312092686459088

We can see that the maximum accuracy of 93.26% comes for the value of generalization constant equal to 100000 and 1000000.

3. **quadratic**

Generalization constant(c)	**Accuracy (in decimal)(1 being highest)(range of 0-1)**
0.01	0.6393917451122375
0.10	0.6567704561911658
1.00	0.6567704561911658
10.00	0.6683562635771181
100.00	0.7139753801593048
1000.00	0.7834902244750181
10000.00	0.837074583635047
100000.00	0.887762490948588
1000000.00	0.9174511223750905
10000000.00	0.9232440260680667
100000000.00	0.9210716871832005

We can see that the maximum accuracy of 92.32% comes for the value of c=10000000.