**Machine Learning Assignment 2**

**Submitted by:**

**18CS10018 Dvij Kalaria**

**18CS10037 Nikhil Popli**

**Task:** Implement Support Vector Machine (SVM), Multilayer Layer Perceptron (MLP) algorithm using sklearn python library and observe variations in accuracy of prediction by changing model parameters like kernel, generalization constant (C) for SVM and learning rate, no of hidden layers, nodes on each layer for MLP.

**Dataset:**The dataset is available at <https://archive.ics.uci.edu/ml/datasets/Spambase>. The dataset consists of 57 features as following:-

48 continuous real [0,100] attributes of type word\_freq\_WORD  
= percentage of words in the e-mail that match WORD, i.e. 100 \* (number of times the WORD appears in the e-mail) / total number of words in e-mail. A "word" in this case is any string of alphanumeric characters bounded by non-alphanumeric characters or end-of-string.  
  
6 continuous real [0,100] attributes of type char\_freq\_CHAR]  
= percentage of characters in the e-mail that match CHAR, i.e. 100 \* (number of CHAR occurences) / total characters in e-mail  
  
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  
= total number of capital letters in the e-mail

And 1 target attribute as Boolean variable whether the mail is spam or not

**Data Pre-processing:**

After analysing the data, it was found out that as the range of values for each attribute is different, it took a lot of time for the SVM and MLP classifiers to converge their parameters. After standardizing the values for each of the attributes, the classification accuracy as well as time of convergence drastically improved. Hence each data value was scaled down according to the range of the attribute it belongs to such that all of them have mean 0 and variance 1 as per the following statistics:-

No Min: Max: Average: Std.Dev: Coeff.Var\_%:

1 0 4.54 0.10455 0.30536 292

2 0 14.28 0.21301 1.2906 606

3 0 5.1 0.28066 0.50414 180

4 0 42.81 0.065425 1.3952 2130

5 0 10 0.31222 0.67251 215

6 0 5.88 0.095901 0.27382 286

7 0 7.27 0.11421 0.39144 343

8 0 11.11 0.10529 0.40107 381

9 0 5.26 0.090067 0.27862 309

10 0 18.18 0.23941 0.64476 269

11 0 2.61 0.059824 0.20154 337

12 0 9.67 0.5417 0.8617 159

13 0 5.55 0.09393 0.30104 320

14 0 10 0.058626 0.33518 572

15 0 4.41 0.049205 0.25884 526

16 0 20 0.24885 0.82579 332

17 0 7.14 0.14259 0.44406 311

18 0 9.09 0.18474 0.53112 287

19 0 18.75 1.6621 1.7755 107

20 0 18.18 0.085577 0.50977 596

21 0 11.11 0.80976 1.2008 148

22 0 17.1 0.1212 1.0258 846

23 0 5.45 0.10165 0.35029 345

24 0 12.5 0.094269 0.44264 470

25 0 20.83 0.5495 1.6713 304

26 0 16.66 0.26538 0.88696 334

27 0 33.33 0.7673 3.3673 439

28 0 9.09 0.12484 0.53858 431

29 0 14.28 0.098915 0.59333 600

30 0 5.88 0.10285 0.45668 444

31 0 12.5 0.064753 0.40339 623

32 0 4.76 0.047048 0.32856 698

33 0 18.18 0.097229 0.55591 572

34 0 4.76 0.047835 0.32945 689

35 0 20 0.10541 0.53226 505

36 0 7.69 0.097477 0.40262 413

37 0 6.89 0.13695 0.42345 309

38 0 8.33 0.013201 0.22065 1670

39 0 11.11 0.078629 0.43467 553

40 0 4.76 0.064834 0.34992 540

41 0 7.14 0.043667 0.3612 827

42 0 14.28 0.13234 0.76682 579

43 0 3.57 0.046099 0.22381 486

44 0 20 0.079196 0.62198 785

45 0 21.42 0.30122 1.0117 336

46 0 22.05 0.17982 0.91112 507

47 0 2.17 0.0054445 0.076274 1400

48 0 10 0.031869 0.28573 897

49 0 4.385 0.038575 0.24347 631

50 0 9.752 0.13903 0.27036 194

51 0 4.081 0.016976 0.10939 644

52 0 32.478 0.26907 0.81567 303

53 0 6.003 0.075811 0.24588 324

54 0 19.829 0.044238 0.42934 971

55 1 1102.5 5.1915 31.729 611

56 1 9989 52.173 194.89 374

57 1 15841 283.29 606.35 214

After pre-processing, no of features became 24 with all the encodings

**Statistical Measure:** Since the task is to predict target category for a Boolean target attribute, accuracy is calculated as the percentage ratio of no of correct predictions to the test data size. More formally, given a learning sample *LSi,* target class value *yi, predicted class value ydi,* the percentage accuracy can be formulated as:-

**Algorithm:** SVM and MLP classifier algorithms are used

PART – 1

* Randomly took 80% of the dataset as a training set and the rest as a test set.
* Implemented the binary SVM classifier using Linear, quadratic and radial kernels
* Varied the value of generalization constant C. More the value, there will be lesser boundary for tolerance (inversely proportional) of misclassifications on training set but may yield better test or generalization accuracy.
* Variation of accuracy for different values of C on different kernels is given as :-
  + Linear
  + Quadratic
  + Radial basis function

PART - 2

* In this part also, we randomly took 80% of the total data as training set and the rest as test set, used the sklearn library to implement MLP neural network classifier.
* Varied the model architecture by changing the no of hidden layers, nodes in each layer, hyper parameters such as learning rate
* Accuracy results for different learning rates on all architectures is as follows :-
  + 0 hidden layer
  + 1 hidden layer with 2 nodes
  + 1 hidden layer with 6 nodes
  + 2 hidden layers with 2 and 3 nodes respectively
  + 2 hidden layers with 3 and 2 nodes respectively
* Architecture and hyper parameters for the best architecture are as follows:-
* Execution times for CPU and GPU on each architecture is as follows:-

PART – 3

* Performance comparison for both the models is as follows :-

Complete log file can be found in the test.txt file.