MACHINE LEARNING - CS60050
ASSIGNMENT 3 REPORT
SVM & ANN

- **GROUP 61:**
  G Sai Chaitanya, 18CS30018.
  G Maneesh Kumar, 18CS10020.

- **Dataset given:** https://archive.ics.uci.edu/ml/datasets/Spambase

- **SVM Implementation:**
  - First, the features names from 'spambase.names' (from the given link) are copied into an array, there are a total of 57 features and target class label, which is **'spam'.**
  - The data is first preprocessed using MinMaxScaler(), then the data is split using sklearn 'train_test_split' into 80:20 for training and testing respectively.
  - Then binary SVM classifier is applied using three different kernels, linear, quadratic and radial basis using SVC() from sklearn.svm.
  - Among the three, radial basis function seems to outperform the other two with a **best test accuracy** of **93.59%** observed for a **Generalization Constant = 10**.
- **ANN Implementation:**
  - The same data that is normalized and splitted above is again used to train using MLP classifier.
  - We used the MLPClassifier() from the sklearn.neural_network. We changed the hidden layers and the nodes corresponding to each layer, as mentioned in Part 2a for stochastic gradient descent solver. We also varied the Learning rates for eachmodel.
  - The plots are then drawn for Learning rate(X-axis) (vs) accuracy(Y-axis) for each model, and model(X-axis) (vs) accuracy(Y-axis) for each of the learning rates mentioned.
  - The best accuracy is observed for the architecture **'1 hidden layer with 6 nodes'**. We varied the **max_iters**, **batch_size**, **solver**, **activation** parameters to observe the trend of accuracies. We found that for the given data, the accuracy is the best when activation is **'tanh'**. Also, we tried for **'adam'** solver, which gave better accuracy than normal **'sgd'** solver (**'adam'** is optimized version of classical **'sgd'** solver though) As the question clearly mentions to use **'sgd'**, we did that.
  - For the best found MLP model,
    - Number of nodes in the input layer = 57 (number of attributes) (+1 additional bias node);
    - Number of nodes in the output layer = 1 (number of target variables);
    - Number of hidden layers = 1;
    - Number of nodes in the hidden layer 1 = 6;
    - Solver = 'sgd'
    - Activation used for inner layers = 'tanh'

- Learning rate = 0.1;
- Test Accuracy = **92.83%;**
- Batch Size = 1500;
- Number of iterations = 551;
- Number of Epochs = 224;

We have checked with many values of hyperparameters, the above set of values provides a better accuracy over other set of values that we tried. The larger learning rates result in unstable training and tiny rates result in a failure to train. Also, for most of our trails on hyper parameters, MLP classifier with 1 hidden layer with 6 nodes gives the best accuracy for learning rate 0.1.

- **RESULTS (RUN ON GOOGLE COLAB):**
  - **For Part 1 (SVM with different Kernel Functions),**

```
#################################################################
              PART 1 (SUPPORT VECTOR MACHINE)
#################################################################
---------------------------------
         a) Linear Kernel
---------------------------------
|   Index |            C |   Train_Accuracy |   Test_Accuracy |
|---------+--------------+------------------+-----------------|
|       1 |       0.0001 |            61.14 |           58.41 |
|       2 |       0.001  |            61.14 |           58.41 |
|       3 |       0.01   |            61.3  |           57.87 |
|       4 |       0.1    |            84.76 |           84.04 |
|       5 |       1      |            90.14 |           89.36 |
|       6 |      10      |            92.85 |           91.53 |
|       7 |     100      |            93.59 |           91.53 |
|       8 |    1000      |            93.83 |           91.31 |
|       9 |   10000      |            93.51 |           91.21 |
|      10 |  100000      |            93.53 |           91.31 |

++++ For SVM with a linear Kernel,
=> Best Test Accuracy is obtained for C = 10.0
=> For this C, Train Accuracy = 92.85
=> For this C, Test Accuracy  = 91.53
```

```
---------------------------------
         b) Quadratic Kernel
---------------------------------
|   Index |            C |   Train_Accuracy |   Test_Accuracy |
|---------+--------------+------------------+-----------------|
|       1 |       0.0001 |            61.14 |           58.41 |
|       2 |       0.001  |            61.22 |           58.52 |
|       3 |       0.01   |            75.27 |           71.88 |
|       4 |       0.1    |            84.1  |           83.82 |
|       5 |       1      |            89.89 |           89.79 |
|       6 |      10      |            94.65 |           92.51 |
|       7 |     100      |            97.26 |           92.4  |
|       8 |    1000      |            98.51 |           91.75 |
|       9 |   10000      |            99.13 |           91.31 |
|      10 |  100000      |            99.4  |           90.88 |

++++ For SVM with a quadratic Kernel,
=> Best Test Accuracy is obtained for C = 10.0
=> For this C, Train Accuracy 94.65
=> For this C, Test Accuracy 92.51
```

```
---------------------------------
       c) Radial Basis Kernel
---------------------------------
|   Index |          C |   Train_Accuracy |   Test_Accuracy |
|---------+------------+------------------+-----------------|
|       1 |       0.01 |             80.3 |           78.83 |
|       2 |        0.1 |            90.22 |           88.82 |
|       3 |          1 |            94.59 |           93.05 |
|       4 |         10 |            96.96 |           93.59 |
|       5 |        100 |             98.7 |           92.94 |
|       6 |       1000 |            99.46 |           91.97 |
|       7 |      10000 |            99.67 |           90.55 |
|       8 |     100000 |             99.7 |           90.45 |
|       9 |      1e+06 |            99.81 |           90.23 |
|      10 |      1e+07 |            99.35 |           89.58 |

++++ For SVM with a Radial Basis Kernel,
=> Best Test Accuracy is obtained, for C = 10.0
=> For this C, Train data accuracy 96.96
=> For this C, Test data accuracy 93.59
```

- **For Part2(MLP Classifier for the models mentioned),**

```
############################################################
          PART 2 (ARTIFICIAL NEURAL NETWORKS)
############################################################

-----------------------------------------------
 (i)For MLP classifier with 0 hidden layers
-----------------------------------------------
|   Index |   Learning Rate |   Test Accuracy |   No of Iterations |
|---------+-----------------+-----------------+--------------------|
|       1 |          1e-05  |           57.87 |                 12 |
|       2 |          0.0001 |           57.87 |                 12 |
|       3 |          0.001  |           58.09 |                 33 |
|       4 |          0.01   |           81.54 |                918 |
|       5 |          0.1    |           88.49 |                616 |

++++ Best Test accuracy is observed for, Learning rate = 0.1 with the Test accuracy = 88.49


-------------------------------------------------------
 (ii) For MLP classifier, 1 hidden layer with 2 nodes
-------------------------------------------------------
|   Index |   Learning Rate |   Test Accuracy |   No of Iterations |
|---------+-----------------+-----------------+--------------------|
|       1 |          1e-05  |           41.59 |                 12 |
|       2 |          0.0001 |           70.79 |                460 |
|       3 |          0.001  |           82.74 |               1630 |
|       4 |          0.01   |           90.01 |                825 |
|       5 |          0.1    |           91.53 |                440 |

++++ Best Test accuracy is observed for, Learning rate = 0.1 with the Test accuracy = 91.53


-------------------------------------------------------
  (iii)For MLP classifier, 1 hidden layer with 6 nodes
-------------------------------------------------------
|   Index |   Learning Rate |   Test Accuracy |   No of Iterations |
|---------+-----------------+-----------------+--------------------|
|       1 |          1e-05  |           48.21 |                 12 |
|       2 |          0.0001 |           50.81 |                 12 |
|       3 |          0.001  |           58.41 |                205 |
|       4 |          0.01   |           90.01 |                845 |
|       5 |          0.1    |           92.83 |                551 |

++++ Best Test accuracy is observed for, Learning rate = 0.1 with the Test accuracy = 92.83
```

(iv)For MLP classifier, 2 hidden layers with 2 and 3 nodes respectively

Index	Learning Rate	Test Accuracy	No of Iterations
1	1e-05	58.41	12
2	0.0001	58.41	12
3	0.001	58.41	59
4	0.01	91.21	723
5	0.1	91.1	144

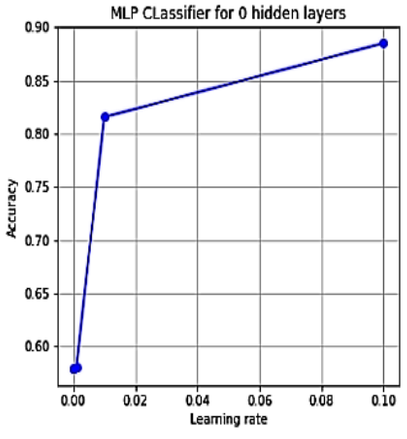
++++ Best Test accuracy is observed for, Learning rate = 0.01 with the Test accuracy = 91.21

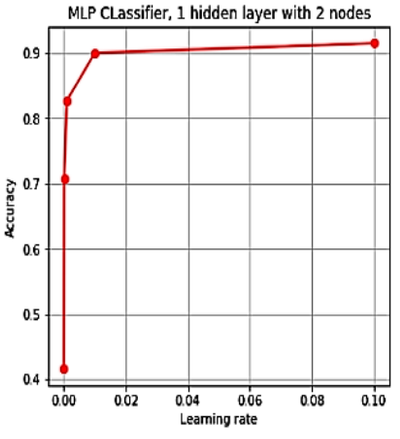
(v)For MLP classifier, 2 hidden layers with 3 and 2 nodes respectively

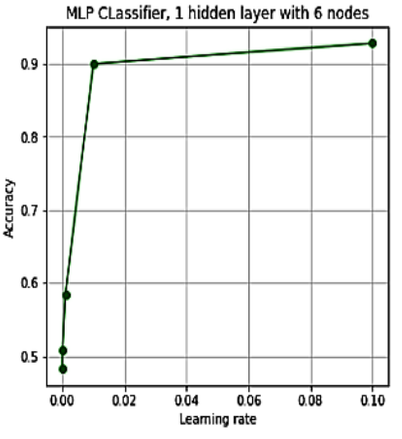
Index	Learning Rate	Test Accuracy	No of Iterations
1	1e-05	41.59	12
2	0.0001	53.85	204
3	0.001	58.31	90
4	0.01	90.99	747
5	0.1	91.31	127

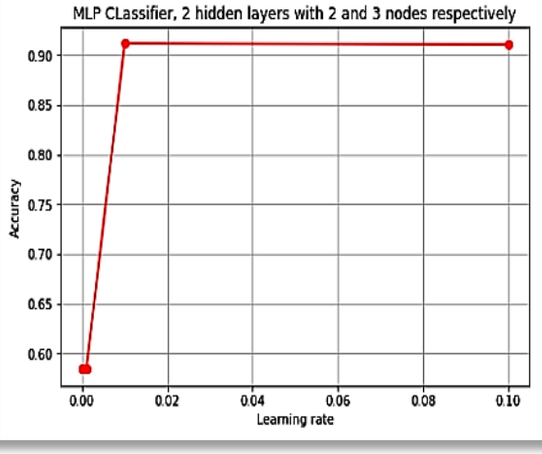
++++ Best Test accuracy is observed for, Learning rate = 0.1 with the Test accuracy = 91.31

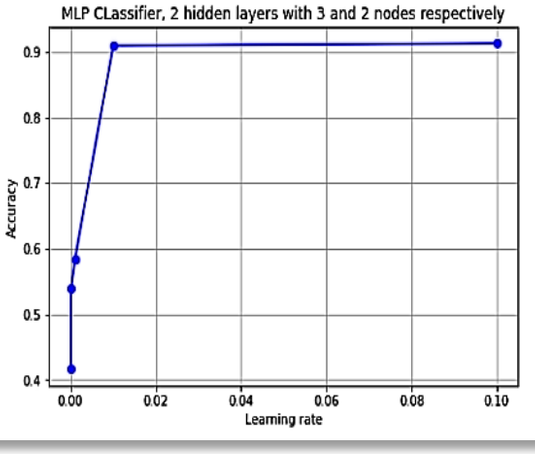
Learning rate vs Accuracy Graphs for each model

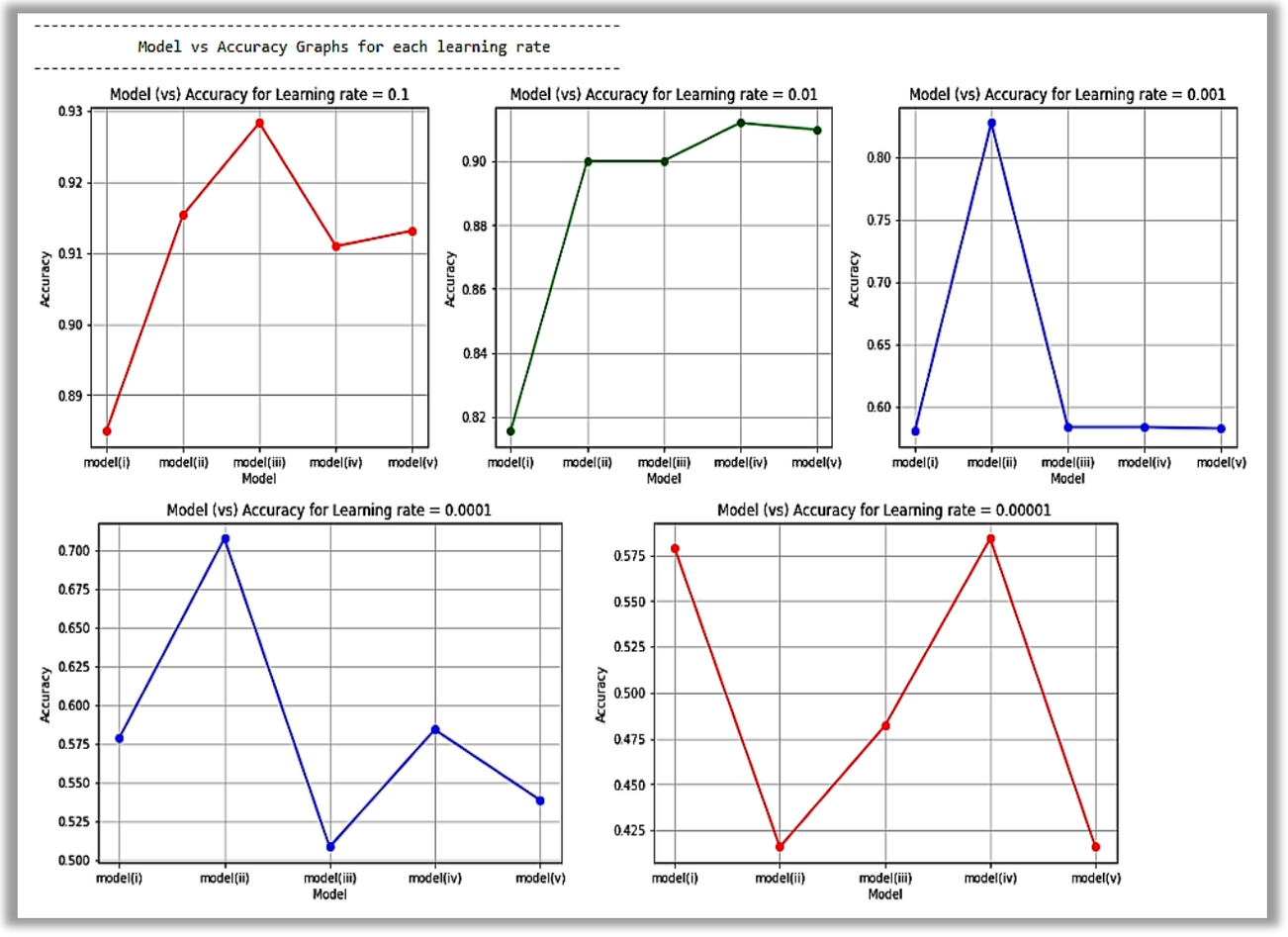













```
----------------------------------------
         Best MLP Model observed
----------------------------------------

++++ MLP CLassifier, 1 hidden layer with 6 nodes

=> Number of nodes in the input layer = 57

=> Number of nodes in the output layer = 1

=> For Learning Rate = 0.1

=> With Test Accuracy = 92.83

=> Loss Computed = 0.1889200467702065

=> Number of Iterations = 551

=> Batch Size = 1500

=> Number of Epochs = 224.0
```

NOTE: Here, model(i) – 0 hidden layers
model(ii) – 1 hidden layer with 2 nodes
model(iii) – 1 hidden layer with 6 nodes
model (iv) - 2 hidden layers with 2 and 3 nodes respectively
model (v) - 2 hidden layers with 3 and 2 nodes respectively

3. Comparision between two models:

For the models that we have trained the SVM with radial basis kernel works better than the other models including the MLP classifier. So SVM outperforms the MLPClassifier.