A Report on Classification

Ekta Katiyar

Person Number: 50291702

November 21, 2018

Abstract

In this experiment, we have performed the task of classification by implementing four different machine learning models. This experiment also helped in understanding various aspects of hyperparameters used in the various classification models. These are the parameters whose value is set before the learning process begins and tuning hyperparameters impacts the Learning process of the models. If hyperparameters are poorly chosen, then the network will learn slowly or might not learn at all and end up affecting the accuracy of the model.

Introduction

Our problem statement was to implement machine learning methods for the task of classification.

Problem Definitions:

1. First, we have to train our logistic regression model backpropagation and tune hyper parameters for MNIST and USPS datasets.
2. Second, train a multilayer perceptron neural network on MNIST and USPS dataset.
3. Third is to train a Random Forest model on MNIST and USPS dataset and tune the hyperparameters.
4. Fourth is to train a Support Vector Machine model on MNIST and USPS dataset and tune the hyperparameters.

Experiments

Below is the list of evaluations done on the implemented models:

1. Evaluate each solution on the test set using classification accuracy using umber of corrected classified data samples and number of samples in the validation set.
2. Construction and evaluation of Strength and weakness of the confusion matrix for each classifier
3. Evaluation of the performance of the ensemble classifier

4. While solving the problem, we shall encounter the "No Free Lunch Theorem". Our objective is to verify that this theorem holds even in this case, as in implementation.

Explanations

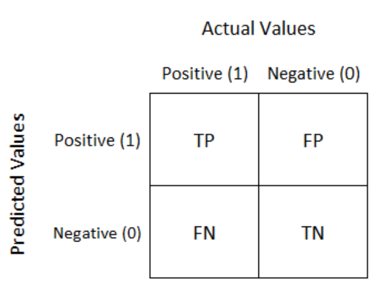
- **USPS Dataset:** USPS dataset consists of 20000 pixels of resolution 100ppi. We have processed the images in the dataset to have same resolution as MNIST dataset images, so that the trained model can be applied to USPS dataset

- **MNIST Dataset:** The MNIST dataset of handwritten digits which consists of 70000 grayscale images, representing 10 digits - 0 to 9. The images are each of 28 x 28-pixel resolution and has a training set of 60,000 examples and test set of 10,000 examples. We need to unpickle the mnist.pkl.gz file using pickle library in order to obtain the training, validation and testing dataset of 50,000, 10,000 and 10,000 examples respectively.
  After retrieving the datasets, we will convert the training target labels against each image in the form of one hot vector.

  We will train our models on MNIST dataset.

- **No Free Lunch Theorem:** The free lunch theorem for search and optimization applies to finite spaces and algorithm that do not resample points. According to no free lunch theorem there is no one model that works best for every problem.
  The assumptions we have considered for one good model may not work equally good for the second model. Therefore, we try out multiple models for our problem and find the best model for the particular problem.
  For our problem statement, the trained model should perform well with MNIST dataset on which they will be trained but would not perform that well on USPS dataset.

- **Confusion matrix:** Confusion Matrix is a performance measurement for machine learning classification model. It is also known as error matrix that allows the visualization of performance of an algorithm. It's a table with 4 different combinations predicted and actual values.



where TP represents True Positive values.
FP represents False Positive values.
FN represents False Negative values.
TN represents True Negative values.
Confusion matrix is extremely useful for measuring recall, precision, specificity, accuracy and AOC-ROC curve.

$$\text{Where, Recall} = \frac{TP}{TP+FN}$$

Out of all positive classes, how much we predicted correctly. It should be high as possible, and

$$\text{Precision} = \frac{TP}{TP+FP}$$

Out of all the classes, how much we predicted correctly. This should also be high.

- **One hot vector:** A one hot encoding is a representation of categorical variables as binary vectors. Each integer value is represented as a binary vector that is all zero values except the index of the integer, which is marked with a 1. Using one hot vector process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

- **Ensemble learning:** Even if we have a hypothesis that is very well suited for a particular problem, it may be very difficult to find the best one. Ensembles generates multiple hypothesis using the same model and tend to yield better results when there is a significant diversity of results among the models.

- **Majority Voting:** Majority voting is also known as hard voting. In this approach, we predict the class label via plurality voting of each classifier. We combine the results from each classifier that classifies the training sample and determine the mode of all the results

- **Bagging:** Stands for "bootstrap aggregating" is an ensemble based algorithm. In this approach, different training data subsets are randomly drawn with replacement from the entire training dataset. Each training data subset is used to train is used to train different classifier of the same type. Results from the individual classifiers will be then combined by taking majority vote.

- **Boosting:** It also creates an ensemble of classifiers by resampling the data which will be later combined by using majority voting. However, in boosting resampling is strategically done to achieve optimal informative training data for each classifier. Boosting is done for three classifiers (say c1, c2, c3) then, first classifier c1 will be trained on the random subset of training data. Second classifier c2 will be trained on only half of the correct data classified by c1 while the third classifier c3 will be trained on the data subset for which c1 and c2 disagreed. A strong classifier in the strict PAC learning sense can then be created by recursive applications of boosting.

- **Softmax function:** Softmax function is used in multiple classification logistic regression model. This function will calculate the probabilities of each target class over all possible target classes which were later used to obtain target class for the given inputs.

IMPLEMENTATION

Multinomial Logistic Regression: Multinomial logistic regression is used to model nominal outcome variables, it is a classification method that generalizes logistic regression to multiclass problems with more than two possible discrete outcomes.

For MNIST dataset: Multinomial Logistic Regression Model-1
Hyperparameters: Learning Rate=0.02, Epochs= 800

<table>
<tr><td>Accuracy</td><td>72.04</td></tr>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.46   0.99   0.63   980
          1   0.98   0.79   0.87   1135
          2   0.89   0.69   0.78   1032
          3   0.66   0.84   0.74   1010
          4   0.98   0.41   0.58   982
          5   0.00   0.00   0.00   892
          6   0.86   0.76   0.80   958
          7   0.97   0.73   0.83   1028
          8   0.44   0.80   0.57   974
          9   0.67   0.72   0.70   1009

avg / total   0.70   0.68   0.66   10000
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix :
[[967   0   1   2   0   0   1   0   9   0]
 [0 893   1 20   0   0   5   0 216   0]
 [117   2 715 67   2   0 33   5 90   1]
 [66   0 14 848   0   0   2   5 72   3]
 [81   0 11 12 401   0 53   0 146 278]
 [426   2 11 210   0   0 11   1 219 12]
 [183   2 10   3   1   0 726   0 33   0]
 [70 13 26 13   1   0   2 748 102 53]
 [75   0   5 91   1   0 12   3 782   5]
 [100   1   9 27   3   0   4 12 125 728]]
</pre></td></tr>
</table>

For USPS dataset: Multinomial Logistic Regression Model-1
Hyperparameters: Learning Rate=0.02, Epochs= 800

<table>
<tr><td>Accuracy</td><td>25.7145</td></tr>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.16   0.75   0.27   2000
          1   0.32   0.06   0.11   2000
          2   0.32   0.38   0.34   1999
          3   0.30   0.40   0.34   2000
          4   0.73   0.18   0.29   2000
          5   0.00   0.00   0.00   2000
          6   0.42   0.12   0.19   2000
          7   0.19   0.08   0.11   2000
          8   0.17   0.27   0.20   2000
          9   0.23   0.03   0.06   2000

avg / total   0.29   0.23   0.19   19999
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix :
[[1495   1 232   48   54   0   17   7   79   67]
 [523 128 225 259   21   0   27 289 512   16]
 [903   9 750   94   5   0   57   19 156   6]
 [902   0   56 807   1   0   18   14 182   20]
 [701   31   77 125 365   0   33 117 474   77]
 [1252   6 135 293   3   0   56   16 228   11]
 [1432   4 188   37   24   0 241   4   68   2]
 [491 128 376 353   1   0   36 155 450   10]
 [1025   5 150 171   15   0   82   7 535   10]
 [454   82 162 494   9   0   9 178 545   67]]
</pre></td></tr>
</table>

For MNIST dataset: Multinomial Logistic Regression Model-2

Hyperparameters: Learning Rate=0.01, Epochs= 1000, Batch size= 50, hidden layer=10

Accuracy	90.82
Classification Report	Classification report: precision recall f1-score support 0 0.92 0.98 0.95 980 1 0.94 0.97 0.96 1135 2 0.91 0.85 0.88 1032 3 0.88 0.89 0.89 1010 4 0.89 0.91 0.90 982 5 0.89 0.81 0.85 892 6 0.92 0.93 0.92 958 7 0.91 0.89 0.90 1028 8 0.85 0.85 0.85 974 9 0.86 0.88 0.87 1009 avg / total 0.90 0.90 0.90 10000
Confusion Matrix	Confusion matrix : [[957 0 2 3 0 0 10 1 7 0] [0 1103 2 4 1 2 4 0 19 0] [13 9 881 19 18 0 20 22 42 8] [6 2 17 898 1 31 7 15 20 13] [2 6 5 0 898 1 10 1 8 51] [16 8 5 46 15 724 19 10 38 11] [16 3 7 2 13 17 895 1 4 0] [3 21 29 4 11 0 0 913 4 43] [9 12 10 31 11 27 13 14 830 17] [14 8 7 10 44 15 0 22 6 883]]

For USPS dataset: Multinomial Logistic Regression Model-2

Hyperparameters: Learning Rate=0.01, Epochs= 1000, Batch size= 50, n_classes=10

Accuracy	33.7952
Classification Report	Classification Report precision recall f1-score support 0 0.29 0.29 0.29 2000 1 0.50 0.14 0.22 2000 2 0.39 0.55 0.46 1999 3 0.29 0.75 0.41 2000 4 0.49 0.42 0.45 2000 5 0.39 0.41 0.40 2000 6 0.60 0.33 0.43 2000 7 0.18 0.28 0.22 2000 8 0.12 0.04 0.06 2000 9 0.19 0.07 0.10 2000 avg / total 0.34 0.33 0.30 19999
Confusion Matrix	Confusion Matrix [[589 1 133 179 146 181 42 454 29 246] [129 288 288 285 274 105 15 490 116 10] [184 37 1097 342 39 122 87 55 14 22] [49 5 130 1491 13 217 11 61 6 17] [78 41 50 84 842 115 38 525 112 115] [154 15 264 489 48 823 78 94 18 17] [272 10 561 192 63 181 663 23 1 34] [169 88 52 818 51 60 10 560 143 49] [382 29 142 660 130 246 147 138 74 52] [51 65 71 686 99 39 12 723 120 134]]

For MNIST dataset: Multinomial Logistic Regression Model-3

Hyperparameters: Random state=1, solver= lbfgs, Multiclass= multinomial

Accuracy	93.07

Classification Report	Classification report: precision recall f1-score support 0 0.95 0.98 0.97 980 1 0.96 0.98 0.97 1135 2 0.93 0.90 0.91 1032 3 0.91 0.91 0.91 1010 4 0.94 0.93 0.93 982 5 0.90 0.87 0.89 892 6 0.93 0.95 0.94 958 7 0.93 0.93 0.93 1028 8 0.88 0.88 0.88 974 9 0.91 0.91 0.91 1009 avg / total 0.93 0.93 0.93 10000
Confusion Matrix	Confusion matrix : [[958 0 0 2 1 7 6 5 1 0] [0 1113 3 1 0 2 4 2 10 0] [4 10 931 16 5 4 16 9 33 4] [4 1 18 918 2 23 4 11 21 8] [1 2 6 3 912 0 9 5 8 36] [10 3 4 36 8 777 13 5 30 6] [9 3 9 1 7 13 912 3 1 0] [1 7 23 9 6 1 0 952 3 26] [8 10 8 19 7 27 14 8 859 14] [10 8 1 9 25 5 0 21 8 922]]

For USPS dataset: Multinomial Logistic Regression Model-3
Hyperparameters: Random state=1, solver= lbfgs, Multiclass= multinomial

Accuracy	36.8723
Classification Report	Classification report: precision recall f1-score support 0 0.40 0.18 0.25 2000 1 0.69 0.13 0.22 2000 2 0.34 0.63 0.45 1999 3 0.25 0.41 0.31 2000 4 0.52 0.34 0.41 2000 5 0.28 0.59 0.38 2000 6 0.65 0.27 0.38 2000 7 0.21 0.36 0.26 2000 8 0.17 0.10 0.12 2000 9 0.28 0.10 0.15 2000 avg / total 0.38 0.31 0.29 19999
Confusion Matrix	Confusion matrix : [[369 1 198 189 79 339 42 502 93 188] [41 264 401 140 254 237 15 489 132 27] [48 29 1267 117 23 336 68 42 46 23] [36 6 310 821 10 679 5 68 46 19] [33 7 81 57 684 203 24 633 166 112] [41 6 353 210 22 1186 33 77 60 12] [84 8 744 91 37 413 543 26 8 46] [78 44 82 597 50 200 11 719 167 52] [179 5 146 533 78 586 84 156 194 39] [14 10 95 549 76 89 6 746 215 200]]

Multilayer Perceptron Neural Network: MLP is a feedforward artificial neural network. It consists of three layer of nodes an input layer, a hidden layer and an output layer. MLP uses a supervised learning approach called Backpropagation.

For MNIST dataset: Multilayer Perceptron Neural Network Model-1
Hyperparameters: Epochs=300, Batch size=50, Hidden layer=2000, learning rate=0.02

Accuracy	83.45

Classification Report

```
Classification report:
             precision    recall  f1-score   support

          0       0.88      0.96      0.92       980
          1       0.85      0.97      0.90      1135
          2       0.87      0.80      0.83      1032
          3       0.75      0.87      0.80      1010
          4       0.82      0.84      0.83       982
          5       0.90      0.55      0.68       892
          6       0.86      0.91      0.89       958
          7       0.87      0.86      0.87      1028
          8       0.79      0.75      0.77       974
          9       0.80      0.78      0.79      1009

avg / total       0.84      0.83      0.83     10000
```

Confusion Matrix

```
Confusion matrix :
[[ 945    0    3    4    0    5   17    1    5    0]
 [    0 1100    5    4    0    0    4    1   21    0]
 [   25   47  823   27   27    0   28   18   37    0]
 [    6    8   29  879    0   14    7   22   36    9]
 [    2   12    4    0  822    0   25    1   12  104]
 [   37   29   12  172   20  488   34   25   53   22]
 [   25    9   19    2   12   13  873    0    5    0]
 [    8   48   21    0   12    0    1  889   12   37]
 [   12   33   21   76   15   17   19   15  735   31]
 [   16   15    6   14   98    6    3   48   12  791]]
```

For USPS dataset: Multilayer Perceptron Neural Network Model-1
Hyperparameters: Epochs=300, Batch size=50, Hidden layer=2000, learning rate=0.02

Accuracy	31.0715

Classification Report

```
Classification report:
             precision    recall  f1-score   support

          0       0.20      0.37      0.26      2000
          1       0.28      0.17      0.21      2000
          2       0.31      0.55      0.40      1999
          3       0.40      0.54      0.46      2000
          4       0.42      0.50      0.45      2000
          5       0.44      0.26      0.33      2000
          6       0.42      0.31      0.36      2000
          7       0.19      0.16      0.17      2000
          8       0.23      0.21      0.22      2000
          9       0.20      0.05      0.08      2000

avg / total       0.31      0.31      0.29     19999
```

Confusion Matrix

```
Confusion matrix :
[[ 731    6  409   57  446   28  112   39   50  122]
 [ 339  330  200  166  233   18   46  397  251   20]
 [ 379   54 1091  103   56   26  129   86   65   10]
 [ 226    8  162 1080   47  157   44   93  135   48]
 [ 217  101   55   43  992   81   47  166  207   91]
 [ 362   29  307  297   55  529  169  101  115   36]
 [ 681   16  383   67  131   28  627   21   39    7]
 [ 253  309  362  323   69   57   56  311  227   33]
 [ 312   49  320  197  153  244  222   41  420   42]
 [ 141  261  212  334  202   36   25  405  281  103]]
```

For MNIST dataset: Multilayer Perceptron Model-2
Hyperparameters: Random State=1, hidden layer size(10,10), alpha=le-5, solver= lbfgs

Accuracy	90.65

<table>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.94   0.97   0.95   980
          1   0.95   0.97   0.96   1135
          2   0.93   0.89   0.91   1032
          3   0.87   0.88   0.87   1010
          4   0.88   0.92   0.90   982
          5   0.82   0.82   0.82   892
          6   0.93   0.93   0.93   958
          7   0.94   0.90   0.92   1028
          8   0.85   0.82   0.83   974
          9   0.88   0.87   0.88   1009

avg / total   0.90   0.90   0.90   10000
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix :
[[953   0   2   1   0   13   9   1   1   0]
 [0 1097   2   5   1   2   2   2   24   0]
 [11   8 917   21   11   9   16   12   24   3]
 [2   7   25 889   1   40   0   18   24   4]
 [3   2   4   0 907   3   9   3   11   40]
 [14   1   4   61   5 733   16   4   40   14]
 [16   5   4   1   11   18 894   1   8   0]
 [5   15   19   14   7   1   0 930   2   35]
 [4   16   7   24   24   63   12   4 798   22]
 [11   3   0   9   67   17   1   12   9 880]]
</pre></td></tr>
</table>

For USPS dataset: Multilayer Perceptron Model-2
Hyperparameters: Random State=1, hidden layer size(10,10), alpha=le-5, solver= lbfgs

<table>
<tr><td>Accuracy</td><td>33.5298</td></tr>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.36   0.28   0.31   2000
          1   0.36   0.14   0.20   2000
          2   0.39   0.51   0.44   1999
          3   0.32   0.53   0.40   2000
          4   0.52   0.42   0.46   2000
          5   0.27   0.47   0.35   2000
          6   0.50   0.39   0.44   2000
          7   0.26   0.34   0.29   2000
          8   0.26   0.12   0.16   2000
          9   0.21   0.17   0.19   2000

avg / total   0.34   0.34   0.32   19999
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix :
[[558   5 123 100 131 260 101 143   68 511]
 [55 282 229   79 293 259   92 562   85   64]
 [141   36 1013 233   42 228 168   76   39   23]
 [44   20 242 1060   5 512   14   65   26   12]
 [25   36   37   28 841 157 105 372   92 307]
 [148   51 187 338   26 943 129   59   54   65]
 [399   8 385 133   50 177 772   16   25   35]
 [60 173 115 497   18 225   18 674   96 124]
 [103   40 183 430   61 563 123 147 235 115]
 [31 134   91 390 163 141   14 518 179 339]]
</pre></td></tr>
</table>

Random Forest: Random forest is an ensemble learning method of classification that involves making of decision trees at the time of training and output. Random forest corrects the overfitting of decision tree on the training dataset.

For MNIST dataset: Random Forest Model-1
Hyperparameters: Number of decision trees=10

Accuracy	94.47

Classification Report

Classification report:

	precision	recall	f1-score	support
0	0.96	0.98	0.97	980
1	0.98	0.99	0.98	1135
2	0.93	0.95	0.94	1032
3	0.91	0.92	0.92	1010
4	0.95	0.95	0.95	982
5	0.93	0.92	0.93	892
6	0.96	0.96	0.96	958
7	0.96	0.94	0.95	1028
8	0.94	0.90	0.92	974
9	0.94	0.92	0.93	1009
avg / total	0.94	0.94	0.94	10000

Confusion Matrix

```
Confusion matrix:
[[ 961    0    3    0    1    5    3    1    5    1]
 [    1 1121    2    4    0    2    2    1    1    1]
 [   10    2  985    6    1    1    7   10    9    1]
 [    1    0   22  934    0   22    2   10   15    4]
 [    0    3    7    1  928    2    7    3    5   26]
 [    5    1    5   37    5  817    8    1   11    2]
 [   11    3    2    1    8    8  921    0    2    2]
 [    0    9   22    5    5    1    0  971    4   11]
 [    7    4    8   25   12   10   10    8  878   12]
 [    8    6    7   18   18    6    0    8    7  931]]
```

For USPS dataset: Random Forest Model-1

Hyperparameters: Number of decision trees=10

Accuracy	31.8215

Classification Report

Classification report:

	precision	recall	f1-score	support
0	0.32	0.29	0.31	2000
1	0.23	0.30	0.26	2000
2	0.29	0.42	0.35	1999
3	0.39	0.50	0.44	2000
4	0.35	0.46	0.40	2000
5	0.32	0.46	0.38	2000
6	0.53	0.25	0.34	2000
7	0.19	0.26	0.22	2000
8	0.33	0.07	0.12	2000
9	0.22	0.06	0.09	2000
avg / total	0.32	0.31	0.29	19999

Confusion Matrix

```
Confusion matrix:
[[ 586   74  300   86  391  160   96  129   21  157]
 [  61  602  130  127  185   69   24  775   18    9]
 [ 217  192  831  122  117  165   78  225   25   27]
 [  86   80  179 1004   91  324   19  165   23   29]
 [  27  291  131  103  919  112   34  295   39   49]
 [ 195  109  185  237  112  911   60  118   29   44]
 [ 386  129  282   84  178  287  505   96   27   26]
 [  77  541  297  241  105  148   22  521   17   31]
 [ 131  215  252  279  206  556   91   78  140   52]
 [  72  355  231  276  334  117   26  387   83  119]]
```

For MNIST dataset: Random Forest Model-2

Hyperparameters: Number of decision trees=100

Accuracy	96.82

Classification Report	Classification report: precision recall f1-score support 0 0.97 0.99 0.98 980 1 0.99 0.99 0.99 1135 2 0.96 0.97 0.97 1032 3 0.95 0.96 0.96 1010 4 0.97 0.97 0.97 982 5 0.97 0.96 0.96 892 6 0.97 0.98 0.98 958 7 0.97 0.97 0.97 1028 8 0.96 0.95 0.95 974 9 0.96 0.95 0.95 1009 avg / total 0.97 0.97 0.97 10000
Confusion Matrix	Confusion matrix: [[969 0 0 0 0 1 4 1 4 1] [0 1118 3 3 1 2 3 0 4 1] [6 1 999 5 2 0 4 9 6 0] [0 0 9 973 0 8 0 9 7 4] [1 0 2 0 957 0 4 0 2 16] [4 0 0 15 3 852 6 2 6 4] [6 3 1 0 2 5 938 0 3 0] [1 3 17 2 3 0 0 994 1 7] [4 0 5 14 3 6 3 5 926 8] [7 6 1 9 12 3 1 4 10 956]]

For USPS dataset: Random Forest Model-2
Hyperparameters: Number of decision trees=100

Accuracy	39.1719
Classification Report	Classification report: precision recall f1-score support 0 0.46 0.32 0.38 2000 1 0.36 0.27 0.31 2000 2 0.42 0.62 0.50 1999 3 0.52 0.63 0.57 2000 4 0.48 0.54 0.50 2000 5 0.33 0.70 0.44 2000 6 0.74 0.37 0.50 2000 7 0.20 0.34 0.25 2000 8 0.52 0.08 0.13 2000 9 0.24 0.05 0.09 2000 avg / total 0.43 0.39 0.37 19999
Confusion Matrix	Confusion matrix: [[649 9 257 63 459 154 72 98 1 238] [31 545 119 117 66 113 23 966 16 4] [89 30 1242 82 57 222 18 250 7 2] [39 9 106 1258 59 345 3 152 3 26] [6 202 67 23 1070 209 21 365 21 16] [148 23 132 108 34 1393 27 118 10 7] [340 56 230 34 93 377 745 111 6 8] [33 323 391 215 43 286 30 674 1 4] [48 53 185 224 119 1032 64 100 154 21] [20 257 240 314 250 139 6 593 77 104]]

For MNIST dataset: Random Forest Model-3
Hyperparameters : Number of decision trees=1000

Accuracy	97.05

<table>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.97   0.99   0.98   980
          1   0.99   0.99   0.99   1135
          2   0.96   0.97   0.97   1032
          3   0.96   0.97   0.96   1010
          4   0.97   0.97   0.97   982
          5   0.97   0.97   0.97   892
          6   0.98   0.98   0.98   958
          7   0.97   0.96   0.97   1028
          8   0.96   0.96   0.96   974
          9   0.96   0.95   0.95   1009

avg / total   0.97   0.97   0.97   10000
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix:
[[969   0   1   0   0   3   3   1   3   0]
 [0 1122   3   3   0   2   2   0   2   1]
 [6   0 999   5   3   0   4   9   6   0]
 [0   0   10 975   0   6   0   9   7   3]
 [1   0   0   0 956   0   5   0   3   17]
 [2   0   0   11   3 861   5   2   5   3]
 [6   3   0   0   3   3 939   0   4   0]
 [1   2   18   1   1   0   0 992   2   11]
 [6   0   5   7   3   5   3   4 931   10]
 [6   5   1   10   12   4   1   5   4 961]]
</pre></td></tr>
</table>

For USPS dataset: Random Forest Model-3

Hyperparameters : Number of decision trees=1000

<table>
<tr><td>Accuracy</td><td>40.82</td></tr>
<tr><td>Classification Report</td><td><pre>
Classification report:
             precision   recall f1-score   support

          0   0.47   0.33   0.39   2000
          1   0.38   0.29   0.33   2000
          2   0.44   0.64   0.52   1999
          3   0.55   0.65   0.59   2000
          4   0.50   0.54   0.52   2000
          5   0.35   0.74   0.47   2000
          6   0.79   0.42   0.55   2000
          7   0.19   0.35   0.25   2000
          8   0.55   0.08   0.14   2000
          9   0.24   0.05   0.08   2000

avg / total   0.45   0.41   0.38   19999
</pre></td></tr>
<tr><td>Confusion Matrix</td><td><pre>
Confusion matrix:
[[650   12 279   53 447 158   56   96   2 247]
 [45 576 111 107   50   93   16 988   13   1]
 [90   29 1289   71   46 190   15 262   5   2]
 [38   7   93 1300   47 308   3 186   3   15]
 [11 203   53   26 1076 182   12 398   19   20]
 [139   29 127   69   22 1473   15 115   7   4]
 [308   50 230   21   83 325 838 135   1   9]
 [37 319 382 232   31 258   32 699   2   8]
 [33   39 150 194   98 1146   67   97 163   13]
 [19 263 222 309 238 127   7 632   82 101]]
</pre></td></tr>
</table>

Support Vector Machine: A Support Vector Machine (SVM) is a discriminative classifier that can be defined by a separating hyperplane. Given a labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

For MNIST dataset: Support Vector Machine model-1

Hyperparameters : Kernel=rbf, c=1

Accuracy	94.35

Classification Report	Classification report:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	980
1	0.97	0.99	0.98	1135
2	0.94	0.93	0.94	1032
3	0.93	0.94	0.93	1010
4	0.93	0.95	0.94	982
5	0.93	0.91	0.92	892
6	0.95	0.96	0.96	958
7	0.95	0.93	0.94	1028
8	0.94	0.91	0.93	974
9	0.94	0.91	0.93	1009
avg / total	0.94	0.94	0.94	10000

Confusion Matrix	Confusion matrix:

```
[[ 967    0    1    0    0    5    4    1    2    0]
 [   0 1120    2    3    0    1    3    1    5    0]
 [   9    1  962    7   10    1   13   11   16    2]
 [   1    1   14  950    1   17    1   10   11    4]
 [   1    1    7    0  937    0    7    2    2   25]
 [   7    4    5   33    7  808   11    2   10    5]
 [  10    3    4    1    5   10  924    0    1    0]
 [   2   13   22    5    7    1    0  954    4   20]
 [   4    6    6   14    8   24   10    8  891    3]
 [  10    6    0   12   33    5    1   14    6  922]]
```

For USPS dataset: Support Vector Machine model-1

Hyperparameters : Kernel=rbf, c=1, gamma= default

Accuracy	38.5419
Classification Report	Classification report:

	precision	recall	f1-score	support
0	0.42	0.29	0.34	2000
1	0.45	0.21	0.29	2000
2	0.35	0.70	0.46	1999
3	0.51	0.56	0.53	2000
4	0.52	0.58	0.55	2000
5	0.29	0.68	0.41	2000
6	0.66	0.37	0.48	2000
7	0.24	0.23	0.23	2000
8	0.37	0.12	0.18	2000
9	0.27	0.10	0.15	2000
avg / total	0.41	0.39	0.36	19999

Confusion Matrix	Confusion matrix:

```
[[ 573    2  428   19  285  248   73   44    6  322]
 [ 110  429  285  137  273  180   46  501   22   17]
 [ 128   18 1402   59   39  198   61   57   23   14]
 [  76    3  186 1123   11  483    5   70   27   16]
 [  18   67   91   14 1167  267   22  194   69   91]
 [ 108   17  257  102   25 1367   60   43   15    6]
 [ 197    7  489   24   98  394  748   13    7   23]
 [  50  225  457  265   57  416   15  452   41   22]
 [  73   25  209  193   87 1006   95   41  244   27]
 [  26  166  228  278  213  165    8  499  214  203]]
```

For MNIST dataset: Support Vector Machine model-2

Hyperparameters : Kernel=linear, c=default, gamma= default

Accuracy	93.89

Classification Report	Classification report:

	precision	recall	f1-score	support
0	0.95	0.98	0.96	980
1	0.97	0.99	0.98	1135
2	0.92	0.94	0.93	1032
3	0.90	0.93	0.92	1010
4	0.93	0.96	0.95	982
5	0.92	0.89	0.91	892
6	0.96	0.95	0.95	958
7	0.95	0.93	0.94	1028
8	0.93	0.89	0.91	974
9	0.95	0.91	0.93	1009
avg / total	0.94	0.94	0.94	10000

Confusion Matrix	Confusion matrix:

```
[[ 959    0    5    2    2    4    7    0    1    0]
 [    0 1121    3    3    0    1    2    1    4    0]
 [    6    8  968    9    3    2   11   10   13    2]
 [    5    2   17  944    4   13    1    8   13    3]
 [    2    1   10    1  943    0    4    2    2   17]
 [   13    4    2   39    5  792    9    1   22    5]
 [   10    3   11    1    5   14  911    2    1    0]
 [    1    8   20   10    6    1    0  961    3   18]
 [    8    4    9   25   11   27    6    5  871    8]
 [    7    6    2   13   32    4    0   18    7  920]]
```

For USPS dataset: Support Vector Machine model-2
Hyperparameters : Kernel=linear, c=default, gamma= default

Accuracy	29.1264
Classification Report	Classification report:

	precision	recall	f1-score	support
0	0.36	0.17	0.24	2000
1	0.49	0.15	0.23	2000
2	0.25	0.65	0.36	1999
3	0.25	0.45	0.32	2000
4	0.46	0.40	0.43	2000
5	0.24	0.44	0.31	2000
6	0.61	0.23	0.33	2000
7	0.23	0.26	0.24	2000
8	0.25	0.08	0.12	2000
9	0.28	0.08	0.13	2000
avg / total	0.34	0.29	0.27	19999

Confusion Matrix	Confusion matrix:

```
[[ 348    0  476  152  222  345   74  172   10  201]
 [  60  303  534  275  230  172   17  351   37   21]
 [ 139   63 1293  115   33  221   55   45   21   14]
 [  56   58  341  898    8  520    9   45   48   17]
 [  24   24  221   82  800  215   10  464   82   78]
 [  47   25  652  240   41  876   30   35   41   13]
 [ 146   19  903   55   86  264  462   38    2   25]
 [  19   74  201  706   54  294   12  522   84   34]
 [ 100   16  298  449  126  692   82   58  160   19]
 [  18   38  204  588  142  104    8  580  155  163]]
```

MAJORITY VOTING for MNIST dataset

Accuracy	93.42
Classification Report	Classification report:

	precision	recall	f1-score	support
0	0.89	0.99	0.94	980
1	0.97	0.99	0.98	1135
2	0.94	0.92	0.93	1032
3	0.89	0.94	0.92	1010
4	0.94	0.95	0.94	982
5	0.95	0.86	0.90	892
6	0.95	0.96	0.95	958
7	0.94	0.94	0.94	1028
8	0.93	0.89	0.91	974
9	0.95	0.91	0.93	1009
avg / total	0.93	0.93	0.93	10000

Confusion Matrix	Confusion matrix: [[972 0 1 2 0 1 2 1 1 0] [0 1121 2 2 0 1 4 1 4 0] [21 6 946 9 6 0 14 12 18 0] [7 0 20 946 0 8 0 13 14 2] [4 1 4 0 930 0 10 2 6 25] [23 2 3 56 6 766 10 4 19 3] [22 3 3 1 5 6 917 0 1 0] [5 11 20 6 3 0 0 963 3 17] [19 7 6 20 11 18 13 9 865 6] [19 7 1 15 27 3 0 17 4 916]]

MAJORITY VOTING for USPS dataset

Accuracy	31.0465
Classification Report	Classification report: precision recall f1-score support 0 0.20 0.68 0.30 2000 1 0.38 0.16 0.22 2000 2 0.36 0.52 0.42 1999 3 0.36 0.51 0.42 2000 4 0.63 0.38 0.47 2000 5 0.48 0.26 0.34 2000 6 0.56 0.20 0.30 2000 7 0.24 0.16 0.19 2000 8 0.21 0.20 0.20 2000 9 0.26 0.03 0.06 2000 avg / total 0.37 0.31 0.29 19999
Confusion Matrix	Confusion matrix: [[1355 2 242 53 136 24 20 21 43 104] [441 321 251 183 114 38 24 363 263 2] [641 34 1041 76 14 24 62 42 59 6] [570 4 115 1024 5 131 17 20 102 12] [436 65 70 95 759 45 21 183 280 46] [801 17 183 289 11 524 52 27 93 3] [1160 7 280 40 43 22 407 4 36 1] [385 221 361 368 12 38 30 322 260 3] [760 21 184 238 43 228 90 34 391 11] [339 163 179 456 63 19 6 350 360 65]]

Results

1. Multinomial Logistic Regression
   Accuracy on MNIST data - 93.07
   Accuracy on USPS data – 36.87
2. Multilayer Perceptron Neural Network
   Accuracy on MNIST data – 90.65
   Accuracy on USPS data – 33.52
3. Random Forest Classifier
   Accuracy on MNIST data – 97.05
   Accuracy on USPS data – 40.82
4. Support Vector Machine
   Accuracy on MNIST data – 94.35

Accuracy on USPS data – 38.54

5. Majority Voting

   Accuracy on MNIST data – 93.42

   Accuracy on USPS data – 31.04

Conclusion

We have verified the 'No Free Lunch Theorem' by testing the performance of our model on the USPS Dataset, and the model accuracy is very less than what we achieved for the MNIST dataset. Also the confusion matrix for each of the classifier model are provided in the report along with the accuracies we have obtained for both MNIST and USPS datasets.

Refrences

[1] Project 3 description posted on UB learns

[2] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62/

[3] http://www.scholarpedia.org/article/Ensemble_learning/

[4] https://en.wikipedia.org/wiki/Random_forest/

[5] https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory/