PROJECT REPORT

CSE574
INTRO TO MACHINE LEARNING

PROJECT 3

BY
YASH AHUJA
50245092
yashahuj@buffalo.edu

1. Introduction

This is a machine learning class project. The goal of this project is to implement an ensemble of four different classifiers, combine the results, and compare all.

2. Problem and model description

- *Definition:* Our objective is to recognize some handwritten digit images using machine learning algorithms. The classifiers used in this project were:
    - Support Vector Machine (SVM)
    - Random Forest
    - Neural Networks
    - Logistic Regression

- *Data sets:* There were two types of data sets provided: *MNIST* and *USPS data sets*. The images were extracted, preprocessed, and split into training, testing and validation sets for the MNIST. The USPS image data was extracted and just tested on the trained MNIST dataset.

- *Hyper-parameters:*

  For SVM, the hyper-parameters varied were:
    - Gamma value
    - Kernel Type

  For Random Forest, the hyper-parameters tuned were:
    - Number of estimators

  For Logistic Regression, the hyper-parameters varied were:
    - Learning Rate

  For Neural Networks, the hyper-parameters tuned were:
    - Activation Function Type
    - Optimizer Type

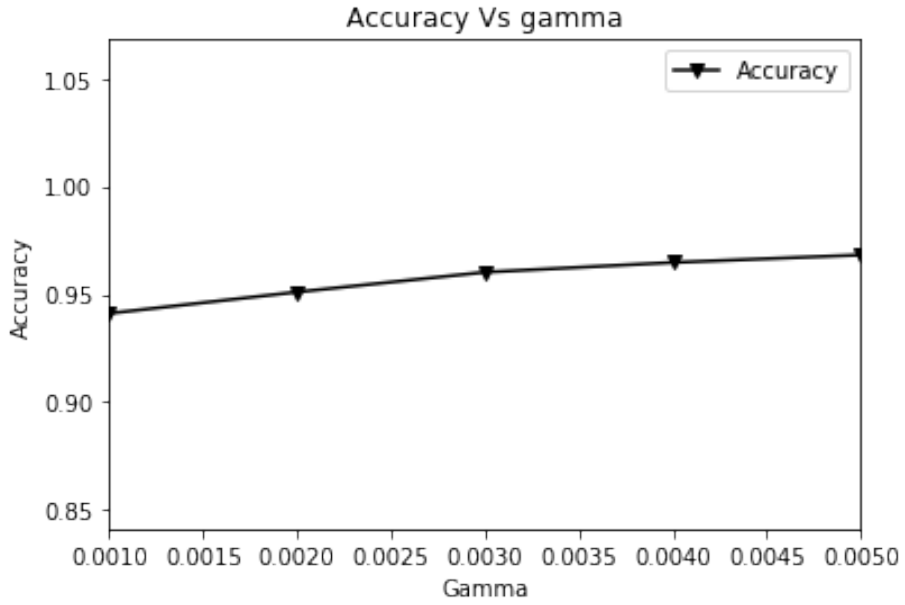
3. Results

PLOTS

SUPPORT VECTOR MACHINE

(i) Hyper-parameter tuning using validation dataset

(a) Tuning hyper-parameter – gamma value

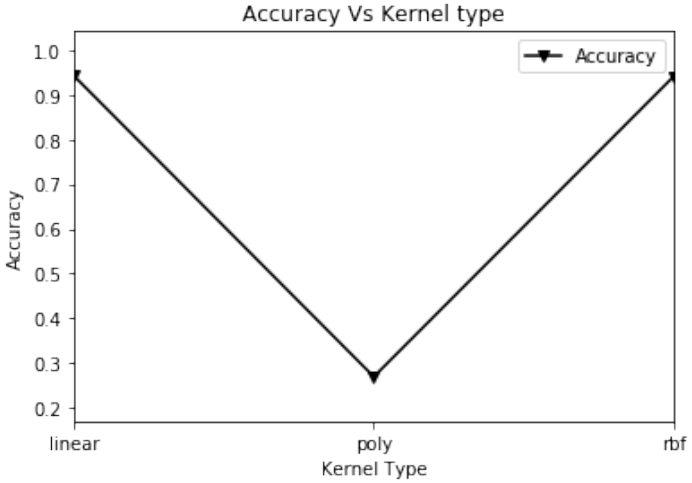


Comments: The hyper-parameter values, gamma, are varied in this graph. As observed, maximum accuracy is obtained at the *gamma value – 0.005*. The **Validation Accuracy** at this value of gamma obtained is **96.82%**

The gamma parameter tells you how far the influence of a training example reaches. If the gamma value is low, the influence of the training example is 'far-fetched'. On the contrary, a high gamma value means the reach of the training example is close.

Note: The gamma value was varied from 0.001 to 0.005 (five values).

(b) Tuning hyper-parameter – kernel type



Comments: The hyper-parameter values, kernel type, are varied in this graph. As observed, the maximum accuracy is obtained at the *kernel – 'rbf'*. The **Validation Accuracy** at this kernel type obtained is **94.11%**

Kernal is a function which maps data to a higher dimension where data is separable.

Note: The kernel types taken into consideration are: 'linear', 'rbf' (radial basis function), 'poly' (polynomial).

(ii) Testing using best values of hyper-parameters

- Optimal gamma value : 0.005
- Optimal kernel type : 'rbf'

Testing results:

S.No	Dataset	Accuracy
1.	MNIST	96.54%
2.	USPS	39.90%

(iii) Confusion Matrix

(a)MNIST dataset

```
Confusion matrix:
[[ 970    0    1    0    0    3    2    1    3    0]
 [   0 1125    3    1    0    1    2    1    2    0]
 [   6    1  991    5    6    0    3    9   11    0]
 [   0    0    6  972    0   10    0   10   10    2]
 [   1    0    6    0  952    0    3    2    2   16]
 [   5    3    0   19    2  845    8    1    7    2]
 [   8    2    2    0    2    4  938    0    2    0]
 [   0   10   17    4    3    0    0  980    1   13]
 [   3    0    4    9    6    8    7    5  930    2]
 [   5    6    4   10   19    2    0    8    4  951]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite good. In the first column, it can be seen that 970 values out of the 998 values are predicted correctly (the totals are not given in this matrix). Similarly, the entire matrix has a strong correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

(a)USPS dataset

```
Confusion matrix:
[[ 603    1  449   23  198  279   55   53    5  334]
 [  88  389  333  164  179  160   44  606   25   12]
 [  93    8 1545   61   23  169   41   45    8    6]
 [  40    2  216 1260    4  419    1   43    8    7]
 [  14   33  148   19 1068  277   22  266   73   80]
 [  85    4  314   92   12 1410   36   32   10    5]
 [ 170    4  568   20   57  366  782   10    2   21]
 [  47  129  562  321   30  376    6  494   20   15]
 [  68    8  281  202   51 1059   64   35  220   12]
 [  14   88  294  324  155  167    5  564  179  210]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite bad. In the first column, it can be seen that only 603 values out of the 1222 values are predicted correctly (the totals are not given

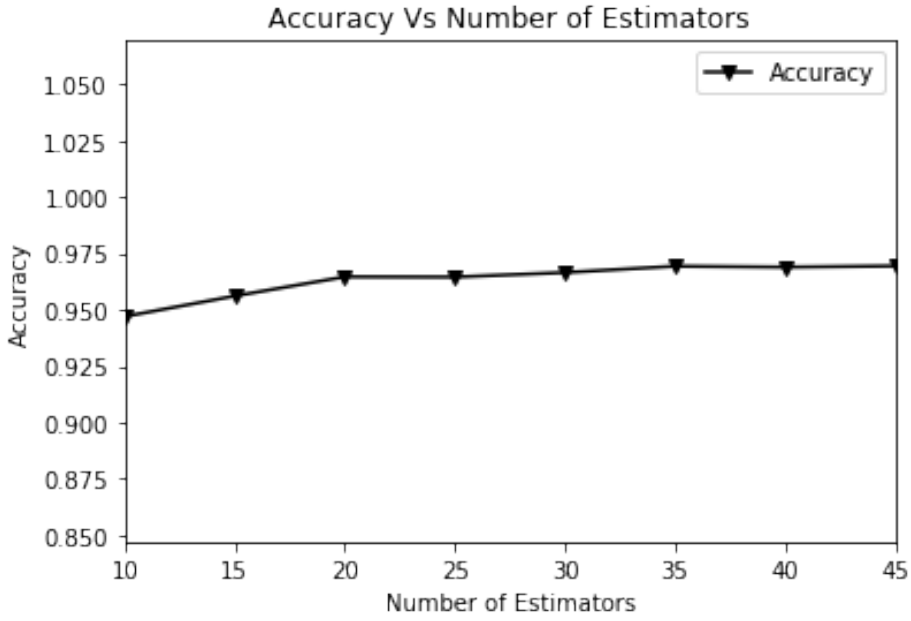
in this matrix). Similarly, the entire matrix has a weak correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

RANDOM FOREST

(i) Hyper-parameter tuning using validation dataset

(a) Tuning hyper-parameter – number of estimators



Comments: The hyper-parameter values, number of estimators, are varied in this graph. As observed, maximum accuracy is obtained at *the number of estimators = 40*. The **Validation Accuracy** at this value of gamma obtained is **97%**

Number of estimators indicates the number of trees. The higher the number of trees, the better is going to be the accuracy. But, after a certain point, there will be saturation, and the result of having more trees will decrease the accuracy.

Note: The number of estimators value was varied from 10 to 45 (eight values).

(ii) Testing using best values of hyper-parameters

- Optimal number of estimators : 45

Testing results:

S.No	Dataset	Accuracy
1.	MNIST	96.7%
2.	USPS	30.58%

(iii) Confusion Matrix

(a)MNIST dataset

```
Confusion matrix:
[[ 972    1    0    0    0    2    3    1    1    0]
 [   0 1125    2    2    0    2    2    0    1    1]
 [   9    0  993    4    3    0    5   10    8    0]
 [   0    0   14  968    0    8    0   10    7    3]
 [   1    0    1    0  957    0    5    0    3   15]
 [   4    3    1   13    3  854    5    1    5    3]
 [   8    3    0    0    5    3  934    0    5    0]
 [   2    1   23    2    0    0    0  987    3   10]
 [   4    0    6    7    7    7    4    6  924    9]
 [   7    5    2   11   12    1    1    5    9  956]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite good. In the first column, it can be seen that 972 values out of the 1005 values are predicted correctly (the totals are not given in this matrix). Similarly, the entire matrix has a strong correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

(a)USPS dataset

```
Confusion matrix:
[[ 689   35  232   83  405  199   77  112   13  155]
 [  62  496  167  110  114   76   37  918    9   11]
 [ 269  135  785   97  139  217   44  266   20   27]
 [ 109   64  164  943  124  340   16  156   27   57]
 [  47  183  107   79  958  161   35  339   49   42]
 [ 217   66  161  230   89 1016   46  139   18   18]
 [ 408   91  296   85  199  318  451  116   16   20]
 [ 137  372  368  227   98  166   44  548   20   20]
 [ 189  119  235  269  216  636   81   97  121   37]
 [  69  272  328  333  237  149   25  398   80  109]]
```

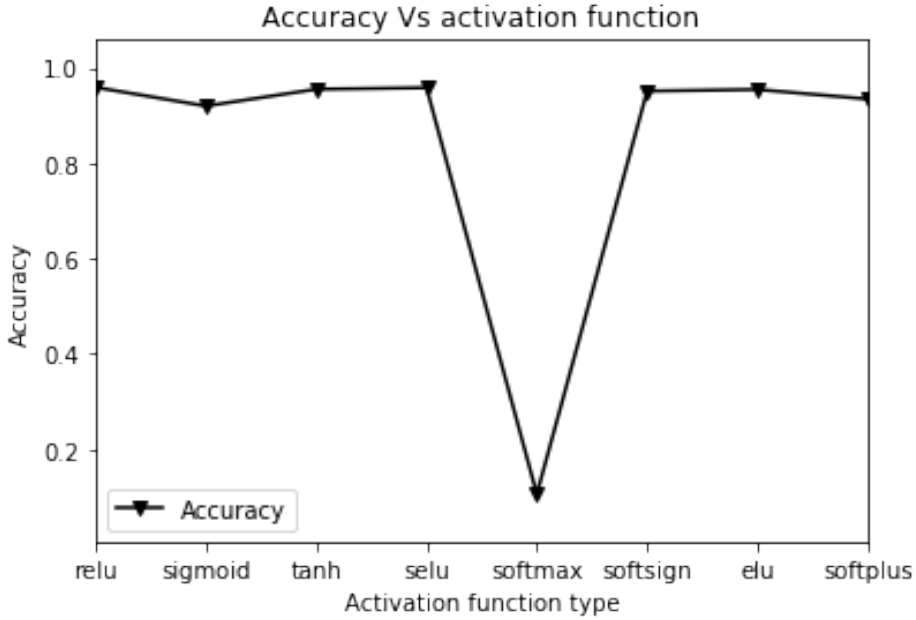
Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite bad. In the first column, it can be seen that only 689 values out of the 2156 values are predicted correctly (the totals are not given in this matrix). Similarly, the entire matrix has an extremely weak correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

NEURAL NETWORKS

(i) Hyper-parameter tuning using validation dataset

(a) Tuning hyper-parameter – activation function type

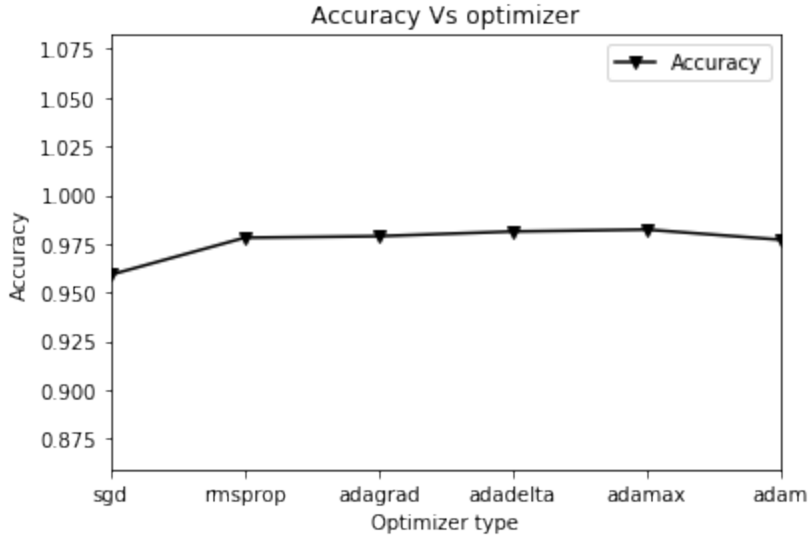


Comments: The hyper-parameter values, activation function type, are varied in this graph. As observed, maximum accuracy is obtained at the activation type '*relu'*. The **Validation Accuracy** at this value of gamma obtained is **95.75%**

'Relu' is more advantageous when compared to the other activation functions is because it does not activate all the neurons at the same time. This is because the input is converted to zero which deactivates neurons in that region, making it more efficient.

Note: The activation types taken into consideration are: 'relu', 'sigmoid', 'tanh', 'selu', 'softmax', 'softsign', 'elu', and 'softplus'.

(b) Tuning hyper-parameter – optimizer type



Comments: The hyper-parameter values, optimizer type, are varied in this graph. As observed, the maximum accuracy is obtained at the *optimizer – 'adadelta'*. The **Validation Accuracy** at this kernel type obtained is **98.11%.**

In my performance, adadelta was gave me the best performance with negligible difference to the others except sgd. Adadelta is the extension of adagrad overcoming the learning decay problem.

Note: The optimizer types taken into consideration are: 'sgd', 'rmsprop', 'adagrad', 'adadelta', 'adamax', 'adam'

(ii) Testing using best values of hyper-parameters

- Optimal activation type : relu
- Optimal kernel type : adadelta

Testing results:

S.No	Dataset	Accuracy
1.	MNIST	97.69%
2.	USPS	47.56%

(iii) Confusion Matrix

(a)MNIST dataset

```
array([[ 968,    1,    1,    1,    3,    0,    3,    1,    2,    0],
       [   0, 1128,    2,    1,    0,    1,    1,    1,    1,    0],
       [   3,    2, 1010,    3,    1,    0,    2,    6,    4,    1],
       [   0,    1,    2,  992,    0,    2,    0,    3,    5,    5],
       [   1,    1,    3,    1,  963,    0,    2,    2,    1,    8],
       [   2,    0,    1,    8,    1,  866,    6,    0,    3,    5],
       [   4,    2,    1,    1,    4,    3,  942,    0,    1,    0],
       [   1,    3,    6,    2,    0,    0,    0, 1010,    1,    5],
       [   3,    0,    3,    6,    3,    4,    3,    2,  946,    4],
       [   3,    2,    0,    3,   10,    6,    0,    5,    2,  978]])
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite good. In the first column, it can be seen that 968 values out of the 985 values are predicted correctly (the totals are not given in this matrix). Similarly, the entire matrix has an extremely strong correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

(a)USPS dataset

```
array([[ 474,   12,  148,  118,  290,  214,  189,  129,   93,  333],
       [  12,  439,  449,  106,  412,  127,   25,  296,   80,   54],
       [  13,    7, 1636,  104,   25,   89,   52,   25,   46,    2],
       [   4,   12,  153, 1426,    5,  294,    5,   13,   80,    8],
       [   6,   87,  105,   17, 1198,  102,   31,  254,  144,   56],
       [  24,    9,  111,  176,   20, 1454,   33,   33,  122,   18],
       [  16,   16,  289,   37,   67,  254, 1135,   60,   64,   62],
       [  13,  134,  413,  468,   53,   36,   21,  669,  179,   14],
       [ 108,   17,  257,  375,   99,  266,  114,   80,  643,   41],
       [   4,   84,  114,  271,  244,   45,    6,  550,  355,  327]])
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite bad. In the first column, it can be seen that only 474 values out of the 674 values are predicted correctly (the totals are not given

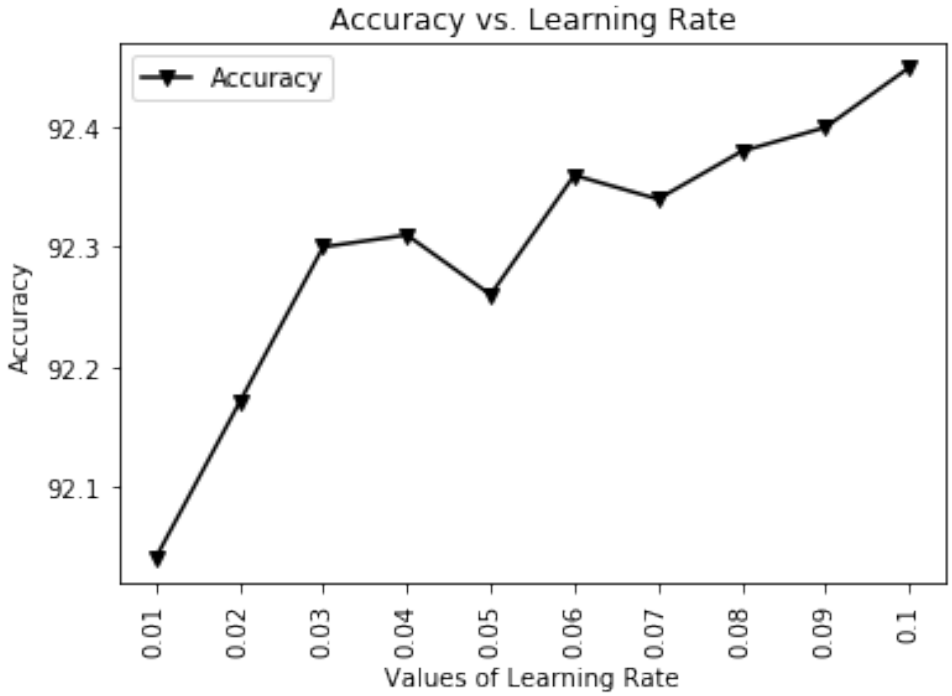
in this matrix). The entire matrix has mixed correlations bringing the total accuracy to around 47%. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

LOGISTIC REGRESSION

(i) Hyper-parameter tuning using validation dataset

(a) Tuning hyper-parameter – learning rate



Comments: The hyper-parameter values, learning rate, are varied in this graph. As observed, maximum accuracy is obtained at *the learning rate= 0.1*. The **Validation Accuracy** at this value of gamma obtained is **92.45%.**
Generally, the lower the learning rate is, the better off it is. This is because we should not overshoot the local minima. In my case, the learning rates taken was considerably low, hence the increase in accuracy w.r.t. learning rate.

Note: The learning rate value was varied from 0.01 to 0.1 (ten values).

(ii) Testing using best values of hyper-parameters

- Optimal learning rate value : 0.1

Testing results:

S.No	Dataset	Accuracy
1.	MNIST	87.53%
2.	USPS	30.83%

(iii) Confusion Matrix

(a)MNIST dataset

```
Confusion matrix:
[[  971     0    22     4     3    23    49     5    19    17  1113]
 [    0  1116    16     1     5     5     3    16    22    12  1196]
 [    1     2   902    19     5     9    21    17     8     0   984]
 [    2     5    29   923     4    50     2    10    41    11  1077]
 [    0     0    13     1   920    10    21     6    16   121  1108]
 [    2     2     5    33     3   741    61     0    68    17   932]
 [    0     1     4     0     0     3   764     0     3     0   775]
 [    3     2    14    12    10    14     2   958    16   116  1147]
 [    1     7    25    13    10    36    35     5   766    22   920]
 [    0     0     2     4    22     1     0    11    15   693   748]
 [  980  1135  1032  1010   982   892   958  1028   974  1009 10000]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite good. In the first column, it can be seen that 971 values out of the 980 values are predicted correctly. Similarly, the entire matrix has a strong correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

(a)USPS dataset

```
Confusion matrix:
[[  533    50   105    53    53    93   298    91   290    26  1592]
 [    3   365    31     8    19     8     5    78    16    26   559]
 [  141   382  1255   304    64   306   678   138   150    90  3508]
 [  147   131   101   725    33   162    87   563   443   467  2859]
 [  116   115    27     8   709    23    65    42   101   101  1307]
 [  287   226   309   734   279  1238   410   207   575   142  4407]
 [   33     9    40     3    19    25   374     6    40     7   556]
 [  480   626    74    95   614    95    25   656   151   809  3625]
 [   99    82    35    49   161    47    16   182   211   232  1114]
 [  161    14    22    21    49     3    42    37    23   100   472]
 [ 2000  2000  1999  2000  2000  2000  2000  2000  2000  2000 19999]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite bad. In the first column, it can be seen that only 533 values out of the 2000 values are predicted correctly. Similarly, the entire matrix has an extremely weak correlation. The diagonals are the values from which we can infer as to how many predictions are correct.

Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

4. Questions to be answered:

(Q1) We test the MNIST trained models on two different test sets: the test set from MNIST and a test set from the USPS data set. Do your results support the "No Free Lunch" theorem?

Solution: When comparing the two datasets, I can say that it supports the theorem. There is no model that works best for every problem. In my case, the MNIST and USPS datasets have very different results. The model was trained on MNIST dataset and even with some assumptions, the USPS accuracy results were very poor.

(Q2) Observe the confusion matrix of each classifier and describe the relative strengths/weaknesses of each classifier. Which classifier has the overall best performance?

Solution: The confusion matrix has been summarized for each of the testing sets for each classifier. When all of the models are compared, it is observed that the overall best performance is given by the neural networks classifier.

(Q3) Combine the results of the individual classifiers using a classifier combination method such as majority voting. Is the overall combined performance better than that of any individual classifier?

Solution:

COMBINED PERFORMANCE – MAJORITY VOTING

(i) Testing results:

S.No	Dataset	Accuracy
1.	MNIST	94.85%
2.	USPS	36.84%

Comments: It is observed that the combined accuracies shown above are not the highest when compared to all the other classifiers. On the contrary, it is not the least too. But, we consider only the best performance. Three out of the four classifiers have better accuracies than the combined model for the MNIST dataset, and two of the four for the USPS dataset. They are as follows:

SVM – 96.54%, RF – 96.7%, NN – 97.69% ----------MNIST
SVM – 39.90%, NN – 47.56% ----------USPS

(ii) Confusion Matrix

(a)MNIST dataset

```
Confusion matrix:
[[  964     0     1     0     0     2     4     0     2     2   975]
 [    0  1111     0     0     0     0     1     0     0     2  1114]
 [    0     2   927     1     1     0     1     4     1     0   937]
 [    0     6    31   925     0     3     0     3     2     2   972]
 [    2     0    11     1   916     0     2     0     1     4   937]
 [    2     1     3    35     3   814     2     0     3     2   865]
 [    4     3     9     0     7    11   910     0     0     1   945]
 [    3     2    17    17    12    14     2   990     2     4  1063]
 [    5     9    31    19    10    41    36     6   938     2  1097]
 [    0     1     2    12    33     7     0    25    25   990  1095]
 [  980  1135  1032  1010   982   892   958  1028   974  1009 10000]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite good. In the first column, it can be seen that 964 values out of the 980 values are predicted correctly. Similarly, the entire matrix has a strong correlation. The diagonals are the values from which we can infer as to how many predictions are correct. This is better than some of the matrices of the other classifiers, but not the best one.
Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

(a)USPS dataset

```
Confusion matrix:
[[  297     4     4     0     0    14     6     3    11     1   340]
 [    1   228     0     0     3     0     0    18     0     1   251]
 [   55   190   875    28    17    26    99    88    26    22  1426]
 [   72    78   108   662     9    39    29   302   110   150  1559]
 [  105    80   101    58   622    11    47    43    53   119  1239]
 [  273   157   365   794   202  1422   279   261   624    97  4474]
 [  172    32   106    16    41    89  1188    43   153    14  1854]
 [  360  1003   297   225   668   197   168   869   186   711  4684]
 [  120   148    88   133   273   161    84   300   741   421  2469]
 [  545    80    55    84   165    41   100    73    96   464  1703]
 [ 2000  2000  1999  2000  2000  2000  2000  2000  2000  2000 19999]]
```

Comments: By looking at this confusion matrix, it can be concluded that the accuracy is quite bad. In the first column, it can be seen that only 297 values out of the 2000 values are predicted correctly. Similarly, the entire matrix has an extremely weak correlation. The diagonals are the values from which we can infer as to how many predictions are correct. This is better than some of the matrices of the other classifiers, but not the best one.
Note: Each row – predicted value (labeled 0-9),
Each column – actual value (labeled 0-9)

5. Conclusion

I applied machine learning classification models on two different datasets. The MNIST dataset was trained, validated and tested on, whereas the USPS dataset was just tested on the trained MNIST dataset. It was found that the best classifier for both the MNIST and USPS was Neural Networks, when compared to the other classifiers. Even after combining and testing the performance, the performance was not as good as the others. The worst of the five was Logistic Regression.