<u>**CSE 574**</u>

<u>**Project 3**</u>

<u>**Report**</u>

Name- Ankit Kumar Sinha
UB Person#-50286874

Task:

Train the following classifier using the Mnist Dataset and testing on both Mnist and USPS dataset:

1. Logistic Regression
2. Neural Network
3. Support Vector Machine
4. Random Forest

Data Preprocessing:

1. The Mnist and USPS images were read from the directory.
2. The target values were fetched from the folder name in case of USPS.
3. Divide the Mnist dataset into 3 parts:
   3.1. Training Data- 80% of the dataset is used for training the model and finding the optimal weights.
   3.2. Validation Data- next 10% of the dataset is used for validation of the model for tuning the hyperparameters on an unseen dataset to achieve the optimal hyperparameters.
   3.3 Testing Data- last 10% of the dataset is used for testing the defined model on unseen data for finding the accuracy and error in it.
4. The Target was converted into one hot Vector form for n class problem.

Dataset:

1. Mnist dataset has 50,000 Training images,10,000 Validation Images, 10,000 Testing Images. It is of size 28*28.
2. USPS dataset has 20,000 Testing images. It was reshaped to 28*28 for maintaining equality with Mnist dataset.

Methodology:

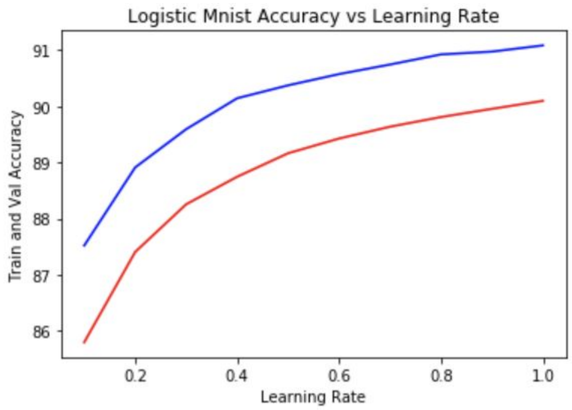
1. Logistic Regression:
   a. In this, the preprocessed training dataset of Mnist was passed in the Logistic model for prediction.
   b. Then, the model was trained, weights were updated and the target was predicted using softmax activation. Here softmax is used since the final output is multiclass.
   c. The updated model was then used for validation on Mnist Validation dataset. In this, the hyperparameters were tuned for optimal performance.
   d. Finally, using the optimal model, Testing was performed on Mnist Test dataset and USPS dataset.
2. Neural Network:
   a. In this, the preprocessed training dataset of Mnist was passed in the Neural network model with one hidden layer for prediction.
   b. Then, the model was trained, weights were updated and the target was predicted by forward and backward propagation using softmax activation at the output layer. Here softmax is used since the final output is multiclass.
   c. The updated model after each epoch is used for validation on Mnist Validation Dataset. In this, the hyperparameters were tuned for optimal performance.
   d. Finally, using the optimal weights, Testing was performed on Mnist Test dataset and USPS dataset.
3. Support Vector Machine:
   a. It is a two-class classifier, so 1vs all mechanism is used to classify for multiclass.
   b. In this, the preprocessed training dataset of Mnist was passed in the SVM model for prediction.
   c. The Trained model was then used for validation on Mnist Validation Dataset. In this, the hyperparameters were tuned for optimal performance.
   d. Finally, using the optimal model, Testing was performed on Mnist Test dataset and USPS dataset.
4. Random Forest:
   a. It is the combination of decision trees. It fits subsample of the dataset in multiple different trees and uses averaging to improve prediction.
   b. In this, the preprocessed training dataset of Mnist was passed in the Random Forest model for prediction.

c. The Trained model was then used for validation on Mnist Validation Dataset. In this, the hyperparameters were tuned for optimal performance.
d. Finally, using the optimal model, Testing was performed on Mnist Test dataset and USPS dataset.

Tuning Hyper Paramerters:

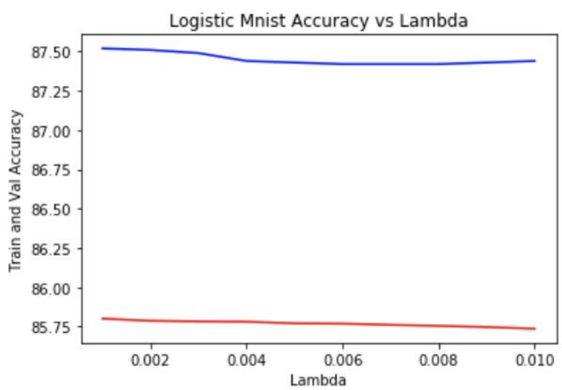
In the Graphs below, Red Line represents Training and Blue Line represents Validation.

1. Logistic Regression:
    a. Learning Rate: This helps the model to converge to local minima.



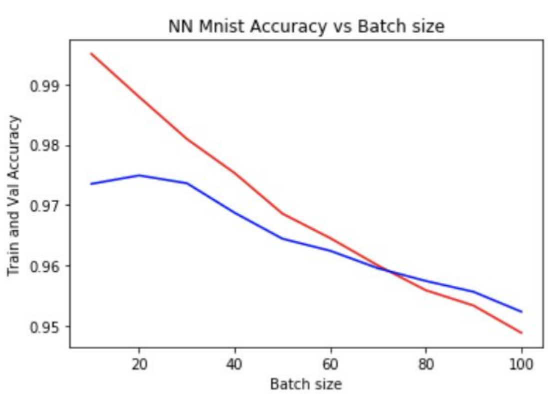
As shown, the accuracy increases with the learning rate but learning rate should not be very high as then it will skip many data points.

b. Lambda:

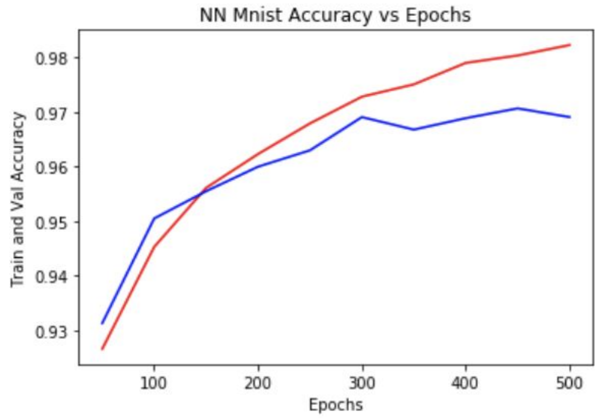


As shown, regularizer is used to prevent overfitting of data and it should not be very high or very low.

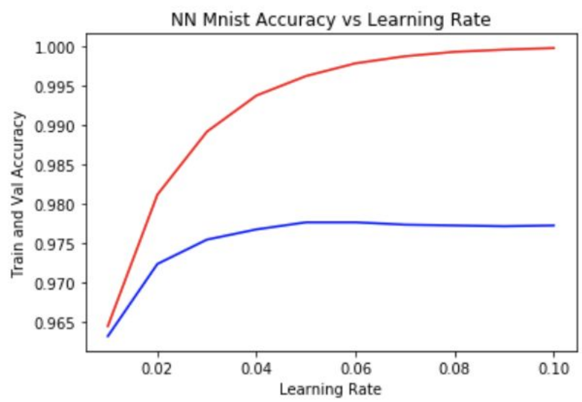
2. Neural Network:
    a. Batch Size: Epochs are divided into small batches of data points. In this, the data points are randomly arranged each time and weights are updated after each batch. A lower number of batches makes the model more accurate since the weights are updated more but also converges more so it should not be very low.



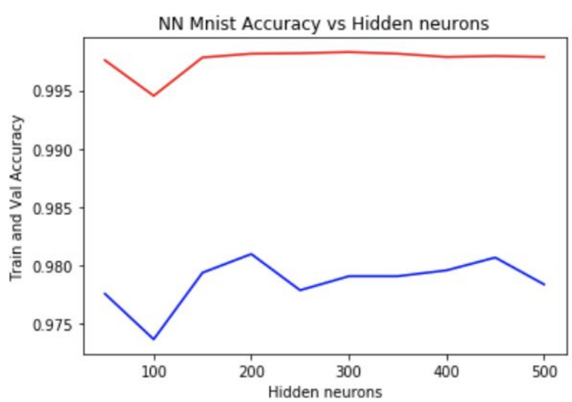
b. Epochs: It is the number of iteration for which the model is trained. The number of epochs must not be very small since the accuracy will be low and it should not be very large as it will converge to be generalized only for the trained dataset.



c. Learning Rate: This helps the model to converge to local minima. If high it will skip many data points and if low will be generalized for the given dataset only.



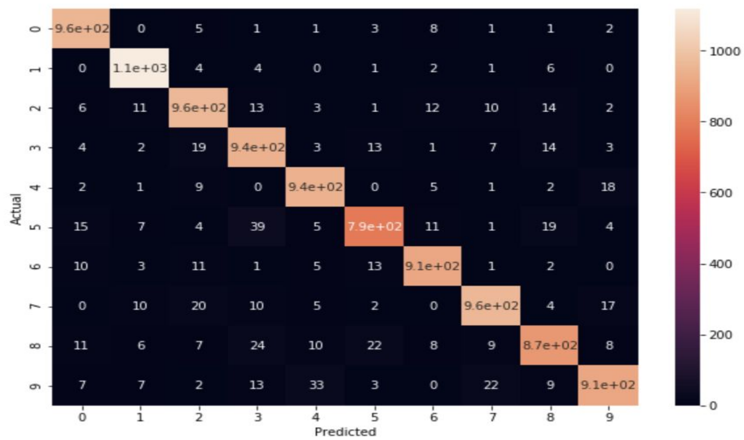
d. Number of Hidden Layers: It is the number of hidden neurons in the hidden layer. High Number of neurons doesn't affect the accuracy much, it becomes constant after some number. This also depends on complexity.



3. Support Vector Machine:
   a. Kernel:
      i. Linear Kernel- It is used where no kernel is needed.

```
Erms: 1.0509995242624994
SVM Linear Mnist Accu: 93.64
```

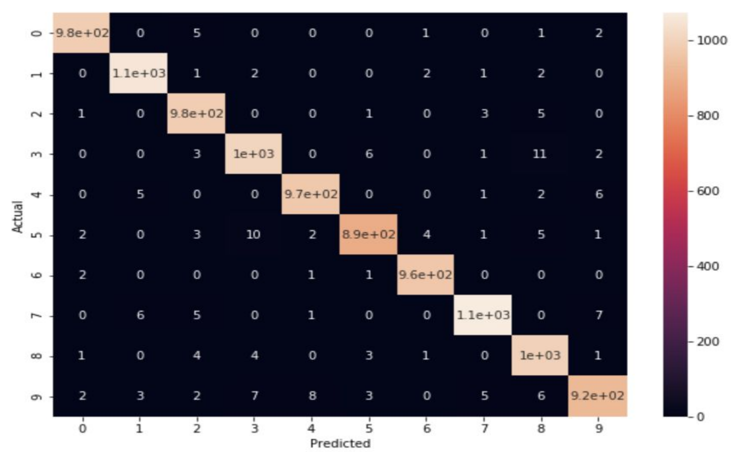
```
Text(69,0.5,'Actual')
```



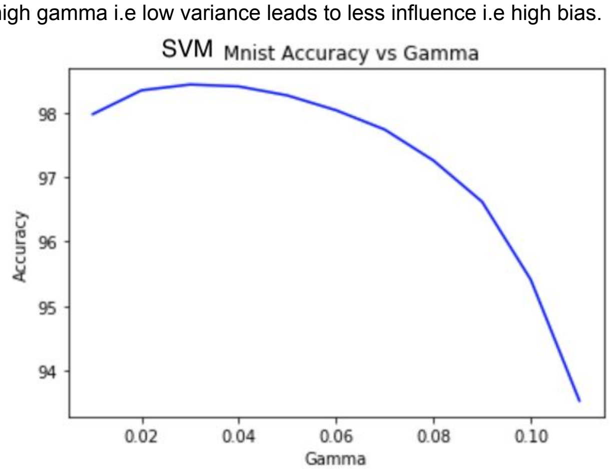
ii. RBF kernel-- In this gram matrix is used. In this, the distance of each data point w.r.t. all data points are calculated.

```
Erms: 0.5770615218501404
SVM Val Accu 98.35
```

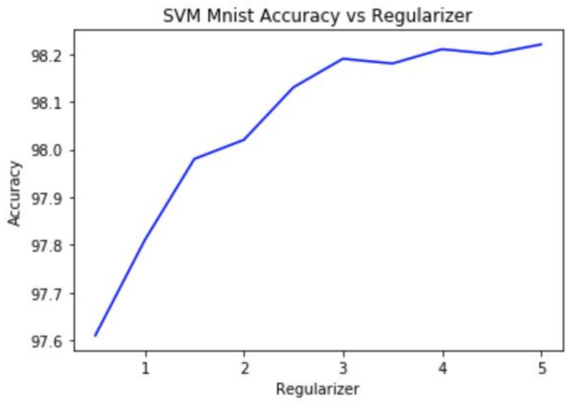
```
Text(69,0.5,'Actual')
```



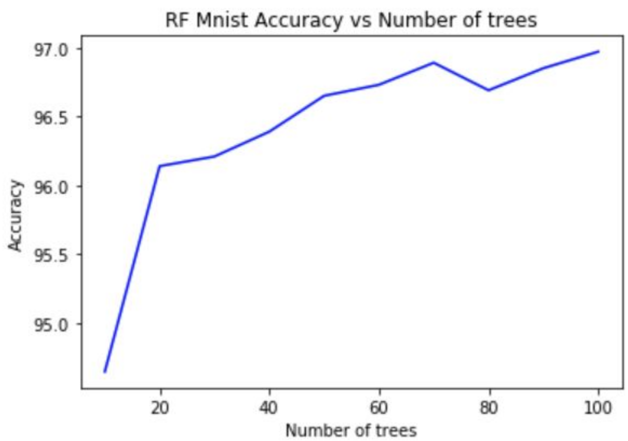
b. Gamma: It is the free parameter of the Gaussian radial basis function. It affects the influence of one class on the other for classification. Low gamma i.e large variance values lead to more influence of one class on other and high gamma i.e low variance leads to less influence i.e high bias.



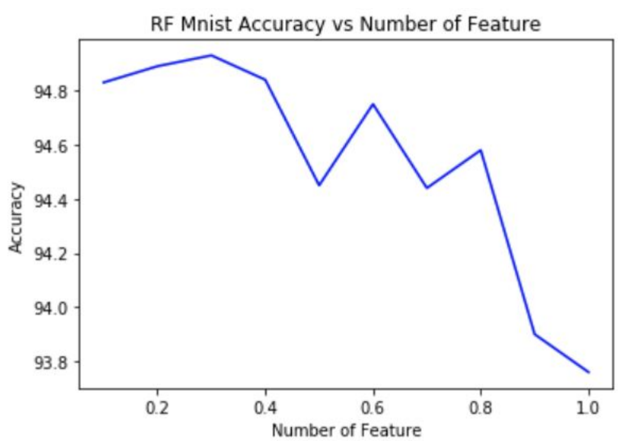
c. C: It is the parameters of soft margin and is used to regularize the data points. It controls the influence of each individual support vector.



4. Random Forest:
    a. n_estimators: This represents the number of trees in the model. Higher values of trees make the model more stable and accurate.



    b. Number of Features: Increasing the number of features will increase the accuracy but a high value can decrease the uniqueness of each tree.

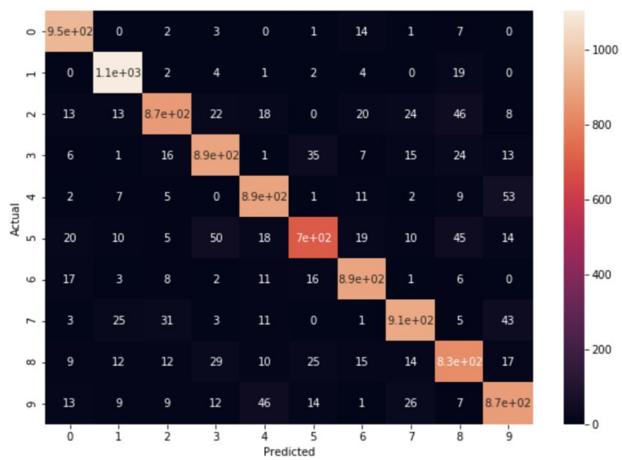


Result:

In Confusion Matrix, the diagonal element represents correct prediction and rest elements represents the wrong prediction. Colour is the measure of the number.

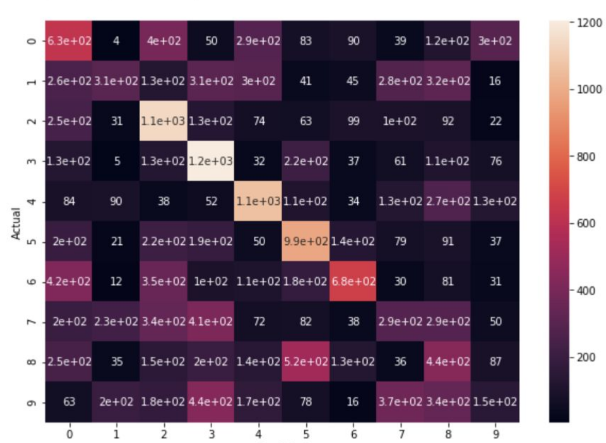
1. Logistic Regression:
    a. Mnist: In this, we can say that the model highest weakness was when actual was 2 and 9 for prediction 8 and 4 respectively. The strongest point for the model is when actual and predicted is both 1.

```
Validation Erms:1.417321417322126
Logisctic Validation Accuracy:89.45
Testing Erms:1.4135062787267696
Logistic Mnist Testing Accuracy:89.11
[[ 952    0    2    3    0    1   14    1    7    0]
 [    0 1103    2    4    1    2    4    0   19    0]
 [   13   13  868   22   18    0   20   24   46    8]
 [    6    1   16  892    1   35    7   15   24   13]
 [    2    7    5    0  892    1   11    2    9   53]
 [   20   10    5   50   18  701   19   10   45   14]
 [   17    3    8    2   11   16  894    1    6    0]
 [    3   25   31    3   11    0    1  906    5   43]
 [    9   12   12   29   10   25   15   14  831   17]
 [   13    9    9   12   46   14    1   26    7  872]]

Text(69,0.5,'Actual')
```



    b. USPS: In this, we can say that the model highest weakness was when actual was 6 for prediction 0. The strongest point for the model is when actual and predicted is both 3.

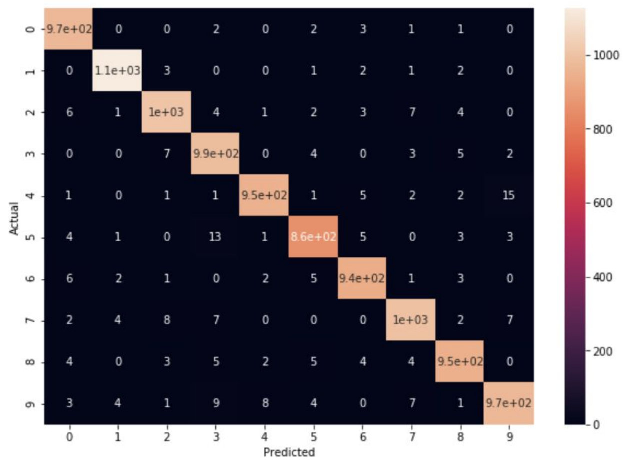
```
USPS Erms: 3.6256940415407564
Logistic USPS Accuracy: 34.47672383619181
[[ 629    4  396   50  294   83   90   39  120  295]
 [ 259  308  126  313  295   41   45  280  317   16]
 [ 246   31 1141  131   74   63   99  100   92   22]
 [ 129    5  128 1203   32  217   37   61  112   76]
 [  84   90   38   52 1063  112   34  127  272  128]
 [ 195   21  218  187   50  987  135   79   91   37]
 [ 424   12  347  100  109  183  683   30   81   31]
 [ 202  227  339  413   72   82   38  291  286   50]
 [ 252   35  153  202  143  521  127   36  444   87]
 [  63  202  180  437  170   78   16  371  337  146]]

Text(69,0.5,'Actual')
```



2. Neural Network:

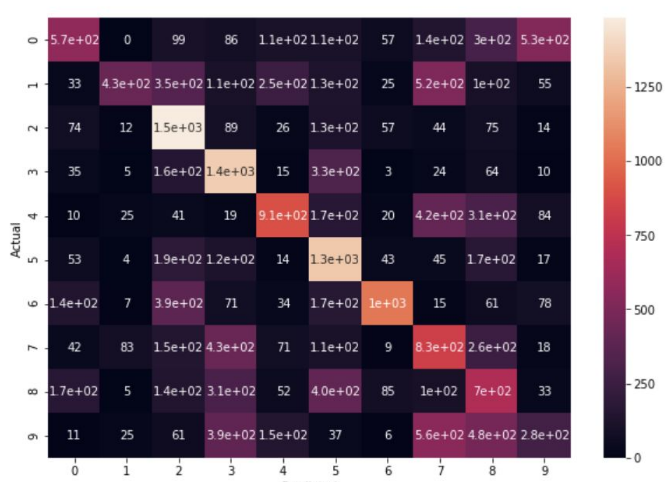
   a. Mnist: In this, we can say that the model highest weakness was when actual was 4 for prediction 9. The strongest point for the model is when actual and predicted is both 1.

```
10000/10000 [==============================] - 0s 19us/step
Loss 0.07426086962409317
NN Mnist Accu: 0.9761
[[ 971    0    0    2    0    2    3    1    1    0]
 [    0 1126    3    0    0    1    2    1    2    0]
 [    6    1 1004    4    1    2    3    7    4    0]
 [    0    0    7  989    0    4    0    3    5    2]
 [    1    0    1    1  954    1    5    2    2   15]
 [    4    1    0   13    1  862    5    0    3    3]
 [    6    2    1    0    2    5  938    1    3    0]
 [    2    4    8    7    0    0    0  998    2    7]
 [    4    0    3    5    2    5    4    4  947    0]
 [    3    4    1    9    8    4    0    7    1  972]]

Text(69,0.5,'Actual')
```



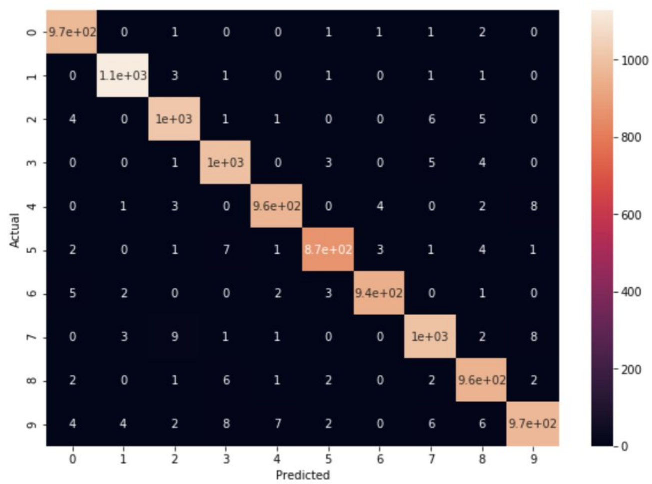
   b. USPS: In this, we can say that the model highest weakness was when actual was 9 for prediction 7. The strongest point for the model is when actual and predicted is both 3.

```
19999/19999 [==============================] - 0s 20us/step
Loss 3.249616170550616
NN USPS Accu: 0.4468223411096049
[[ 574    0   99   86  108  107   57  142  299  528]
 [  33  429  353  108  249  130   25  515  103   55]
 [  74   12 1482   89   26  126   57   44   75   14]
 [  35    5  161 1357   15  326    3   24   64   10]
 [  10   25   41   19  907  168   20  415  311   84]
 [  53    4  190  123   14 1340   43   45  171   17]
 [ 144    7  386   71   34  167 1037   15   61   78]
 [  42   83  147  434   71  108    9  830  258   18]
 [ 168    5  141  310   52  405   85  104  697   33]
 [  11   25   61  394  149   37    6  556  478  283]]

Text(69,0.5,'Actual')
```



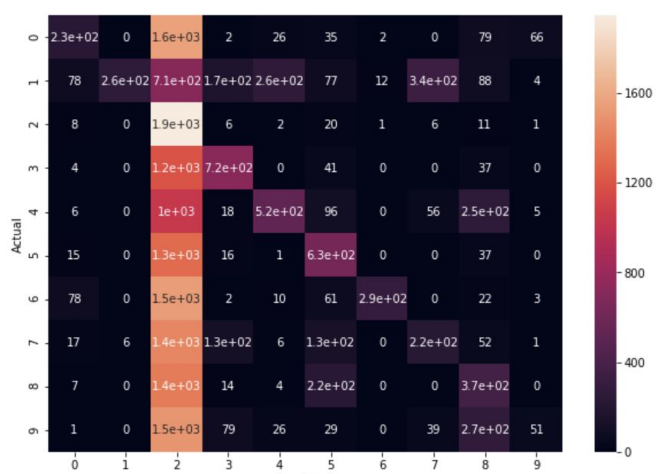
3. Support Vector Machine: In this, we can say that the model highest weakness was when actual was 7 for prediction 2. The strongest point for the model is when actual and predicted is both 1.
    a. Mnist:

```
Erms: 0.5917769850205397
SVM Mnist Accu: 98.27
[[ 974    0    1    0    0    1    1    1    2    0]
 [    0 1128    3    1    0    1    0    1    1    0]
 [    4    0 1015    1    1    0    0    6    5    0]
 [    0    0    1  997    0    3    0    5    4    0]
 [    0    1    3    0  964    0    4    0    2    8]
 [    2    0    1    7    1  872    3    1    4    1]
 [    5    2    0    0    2    3  945    0    1    0]
 [    0    3    9    1    1    0    0 1004    2    8]
 [    2    0    1    6    1    2    0    2  958    2]
 [    4    4    2    8    7    2    0    6    6  970]]

Text(69,0.5,'Actual')
```



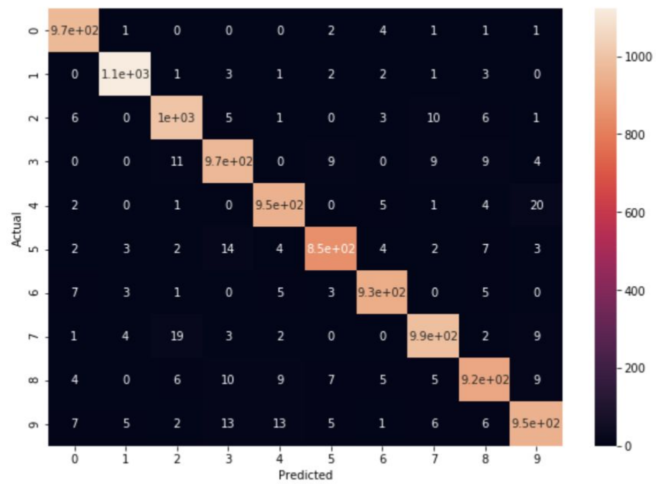
    b. USPS: In this, we can say that the model highest weakness was for prediction 2. The strongest point for the model is when actual and predicted is both 2.

```
Erms: 3.6276863157390373
SVM USPS Accu: 26.141307065353267
[[ 226    0 1564    2   26   35    2    0   79   66]
 [   78  257  713  172  262   77   12  337   88    4]
 [    8    0 1944    6    2   20    1    6   11    1]
 [    4    0 1193  725    0   41    0    0   37    0]
 [    6    0 1045   18  522   96    0   56  252    5]
 [   15    0 1305   16    1  626    0    0   37    0]
 [   78    0 1534    2   10   61  290    0   22    3]
 [   17    6 1435  129    6  134    0  220   52    1]
 [    7    0 1387   14    4  221    0    0  367    0]
 [    1    0 1508   79   26   29    0   39  267   51]]

Text(69,0.5,'Actual')
```



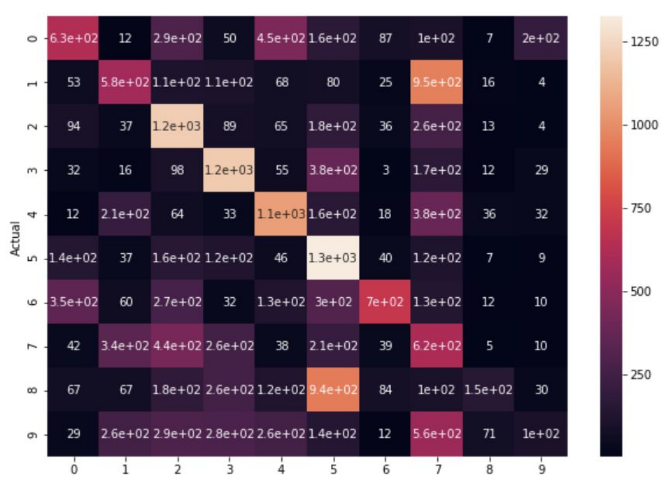
4. Random Forest:
    a. Mnist: In this, we can say that the model highest weakness was when actual was 4 for prediction 9. The strongest point for the model is when actual and predicted is both 1.

```
Erms: 0.8232860985101108
RF Mnist Accu: 96.52
[[ 970    1    0    0    0    2    4    1    1    1]
 [    0 1122    1    3    1    2    2    1    3    0]
 [    6    0 1000    5    1    0    3   10    6    1]
 [    0    0   11  968    0    9    0    9    9    4]
 [    2    0    1    0  949    0    5    1    4   20]
 [    2    3    2   14    4  851    4    2    7    3]
 [    7    3    1    0    5    3  934    0    5    0]
 [    1    4   19    3    2    0    0  988    2    9]
 [    4    0    6   10    9    7    5    5  919    9]
 [    7    5    2   13   13    5    1    6    6  951]]

Text(69,0.5,'Actual')
```



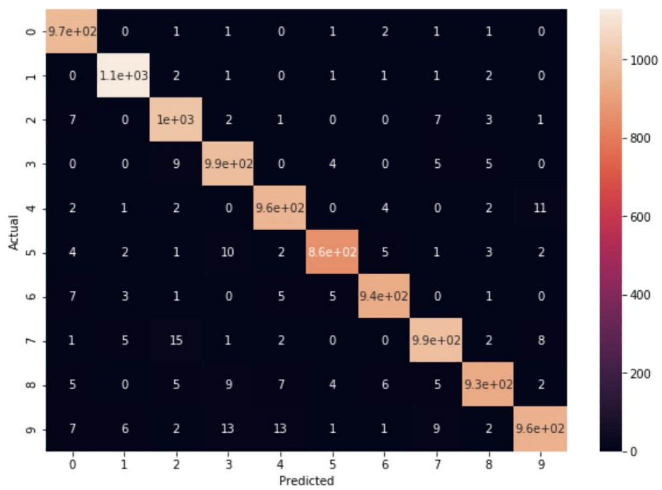
    b. USPS: In this, we can say that the model highest weakness was for prediction 5 and 7. The strongest point for the model is when actual and predicted is both 2.

```
Erms: 3.486718781374979
RF USPS Accu: 37.886894344717234
[[ 633   12  290   50  452  163   87  102    7  204]
 [   53  577  114  109   68   80   25  954   16    4]
 [   94   37 1213   89   65  184   36  264   13    4]
 [   32   16   98 1209   55  379    3  167   12   29]
 [   12  209   64   33 1051  161   18  384   36   32]
 [  138   37  157  122   46 1324   40  120    7    9]
 [  350   60  269   32  126  304  703  134   12   10]
 [   42  342  440  259   38  209   39  616    5   10]
 [   67   67  180  259  124  939   84  101  149   30]
 [   29  255  288  285  256  144   12  558   71  102]]

Text(69,0.5,'Actual')
```



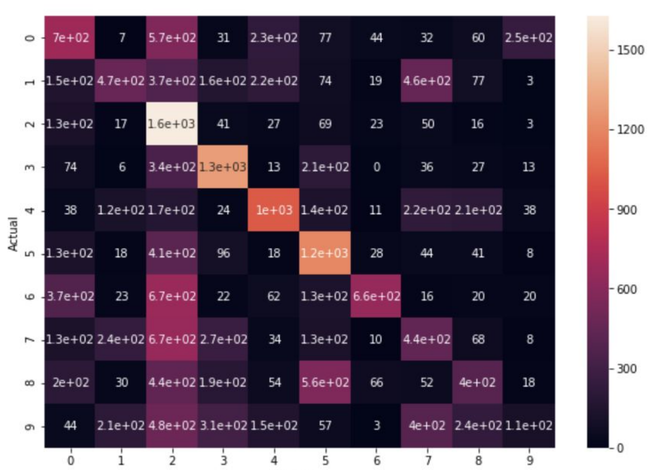
5. Majority Voting:
It is a combination of all the above 4 model and the prediction is the average of those model with the highest voted output.
   a. Mnist: In this, we can say that the model highest weakness was when actual was 7 for prediction 2. The strongest point for the model is when actual and predicted is both 1.

```
Erms: 0.74
Majority Voting Mnist Accu: 97.36
[[ 973    0    1    1    0    1    2    1    1    0]
 [   0 1127    2    1    0    1    1    1    2    0]
 [   7    0 1011    2    1    0    0    7    3    1]
 [   0    0    9  987    0    4    0    5    5    0]
 [   2    1    2    0  960    0    4    0    2   11]
 [   4    2    1   10    2  862    5    1    3    2]
 [   7    3    1    0    5    5  936    0    1    0]
 [   1    5   15    1    2    0    0  994    2    8]
 [   5    0    5    9    7    4    6    5  931    2]
 [   7    6    2   13   13    1    1    9    2  955]]

Text(69,0.5,'Actual')
```



   b. USPS: In this, we can say that the model highest weakness was for prediction 2. The strongest point for the model is when actual and predicted is both 2.

```
Erms: 3.5122808552051032
Majority Voting USPS Accu: 39.63198159907996
[[ 697    7  569   31  232   77   44   32   60  251]
 [ 147  470  367  165  218   74   19  460   77    3]
 [ 127   17 1626   41   27   69   23   50   16    3]
 [  74    6  340 1285   13  206    0   36   27   13]
 [  38  118  174   24 1032  135   11  223  207   38]
 [ 131   18  413   96   18 1203   28   44   41    8]
 [ 370   23  673   22   62  134  660   16   20   20]
 [ 129  242  666  273   34  128   10  442   68    8]
 [ 199   30  437  187   54  560   66   52  397   18]
 [  44  207  485  307  147   57    3  399  237  114]]

Text(69,0.5,'Actual')
```



Conclusion:

1. Accuracy Table for Model:

Model	Mnist Test Data	USPS Test Data
Logistic Regression	89.11	34.47
Neural Network	97.61	44.68
Support Vector Machine	98.27	26.14
Random Forest	96.52	37.88
Majority Voting	97.3	39.63

2. The No Free Lunch Theroem states that no algorithm is best for all generic and best case. In this project, we train the models on Mnist Train Set and test on Mnist test Data and USPS Data. From the above table, we observe that all the model perform well on Mnist Test Set but perform relatively poor on USPS data set since the USPS dataset is differently generate than the Mnist dataset. So my result supports No Free Lunch Theorem.

3. Best Classifier on the basis of the accuracy of both Mnist and USPS Test set can be said to Neural Network Model since it has the least difference between the two test sets.
4. The Combined model (Majority voting) performs better than the Logistic Regression Model, the SVM model and to some extent also Random forest. The major difference was in respect to USPS dataset accuracy.