Project 3 Report

Redwan Ibne Seraj Khan(UBITname: rikhan)
Department of Computer Science and Engineering
University at Buffalo
Buffalo, NY-14228
rikhan@buffalo.edu

Abstract

In this project different machine learning models like logistic regression, multilayer perceptron neural networks, support vector machine and random forest was used to solve a problem of determining the similarity between the samples of the MNIST dataset. After training the machine learning model on MNIST dataset the model was later tested on the USPS dataset. At first an ensemble of four classifiers was implemented and later the results of the individual classifiers were combined to make a final decision.

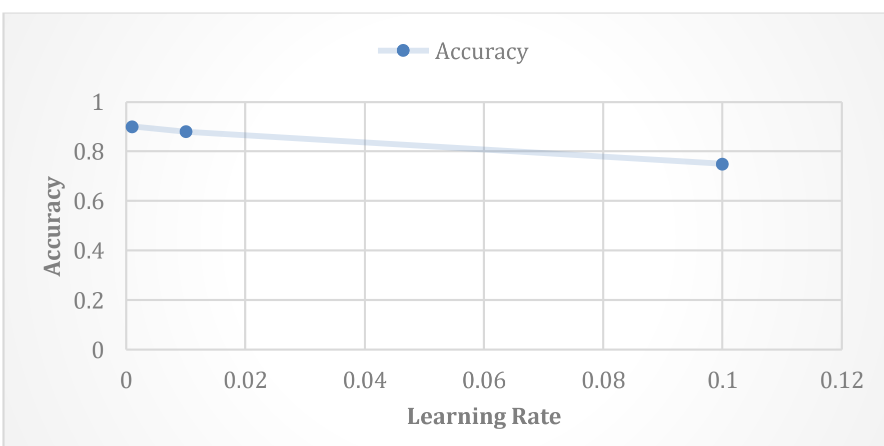
1 Logistic Regression

The softmax function was used in the logistic regression model. This was done as sigmoid function only accounted for binary classification but in MNIST and USPS dataset were have to classify the digits into 10 class. After calculations the loss function was obtained which was used to update the weights by taking the differential of the loss function. The model was tested with various hyperparamters. The best model that was built had iterations of 1000 and learning rate of 0.01. The output from running the model in python notebook is given below.

```
prediction accuracy of MNIST set:0.902
prediction accuracy of USPS set:0.10000500025
```

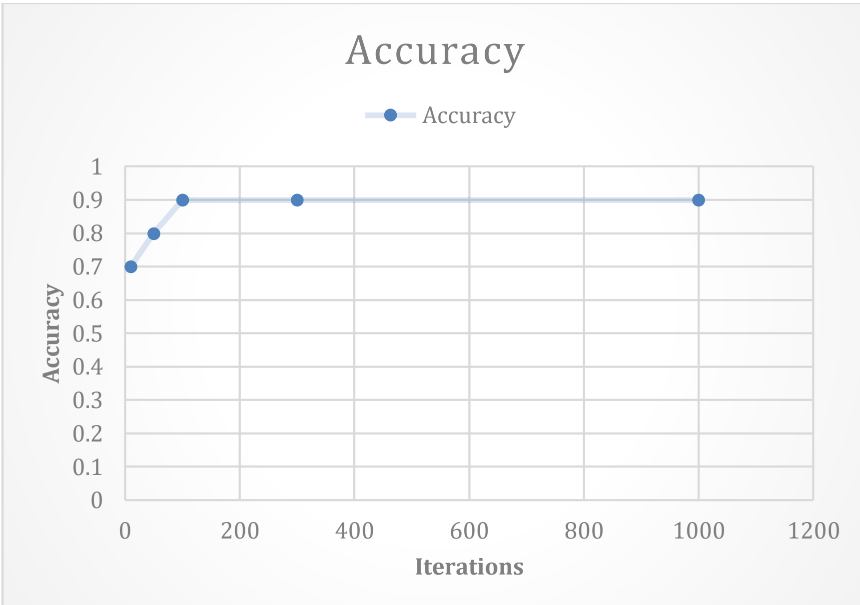
1.1 Learning rate

As the learning rate was decreased the accuracy increased but the time to run the model increased.



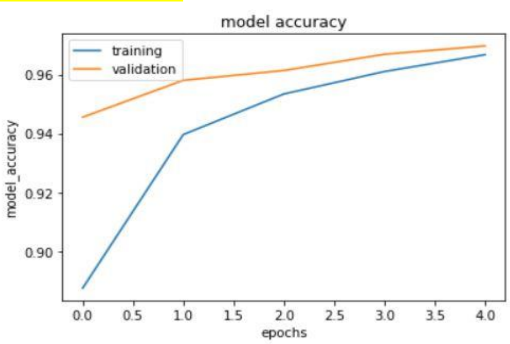
1.2 Number of Iterations

As the number of iterations increased the accuracy increased for some time but then it became concentrated. The time also increased with the increase in iterations.



2 Neural Networks

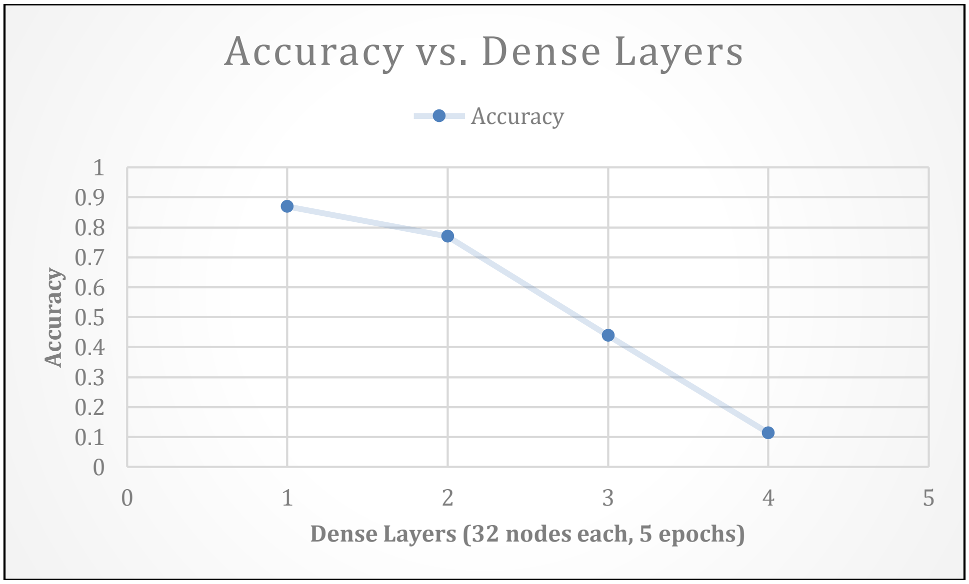
A sequential model was used for neural networks. Four dense layers having a combination of different activation function and nodes was used. The first layer had 1024 nodes and relu activation function. The second layer had 512 nodes and the sigmoid activation function. The third layer had 128 nodes with relu activation function. The last layer had 10 nodes and used softmax activation function.




```
Test loss: 0.1108
Test accuracy: 0.9657
```

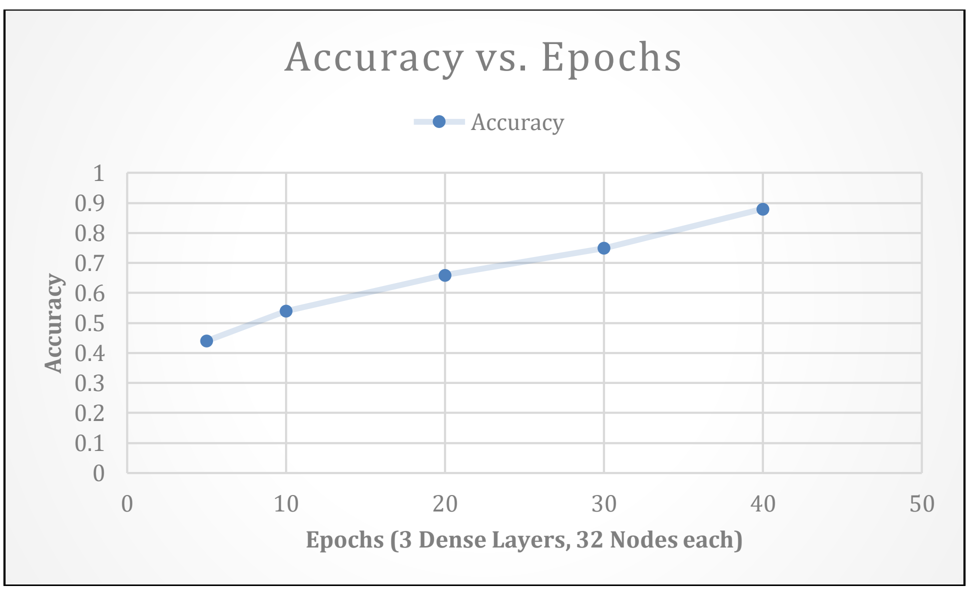
The model was tested with different hyperparameters. The results have been shown below.

2.1 Number of Dense Layers and Activation Functions



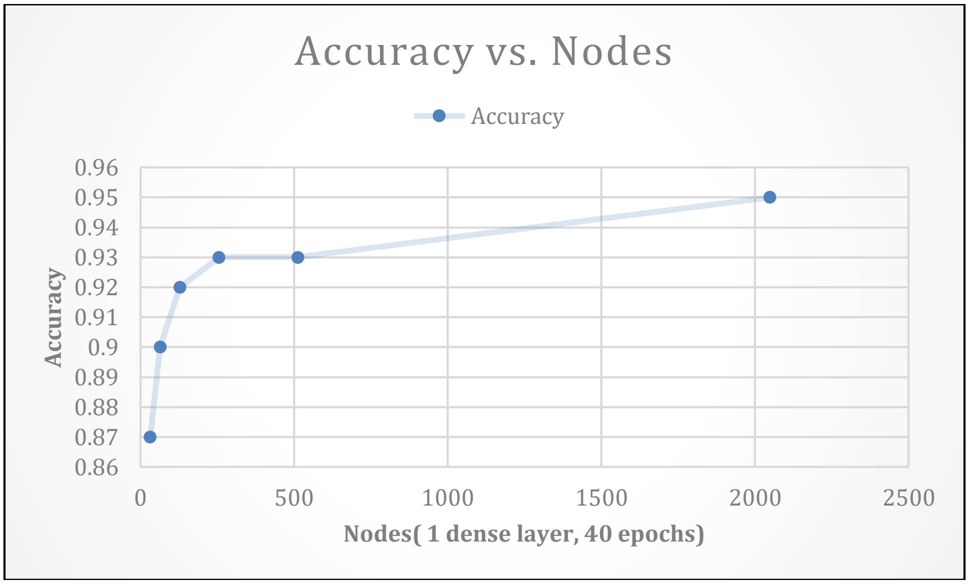
The number of dense layers was gradually increased keeping the number of nodes constant. Sigmoid activation function was used in the dense layers. It was seen that the performance deteriorated when increasing the number of layers keeping the number of nodes constant.

2.2 Number of Epochs



It was seen that the model performed better if the number of epochs was increased. The number of nodes was kept constant at 32 Nodes and 3 dense layers were used at various combinations of activation functions like relu and sigmoid.

2.3 Number of Nodes



The number of dense layers was kept constant at one and the number of epochs was also kept constant at 40. It was seen that as the number of nodes were increased there was a remarkable increase in performance.

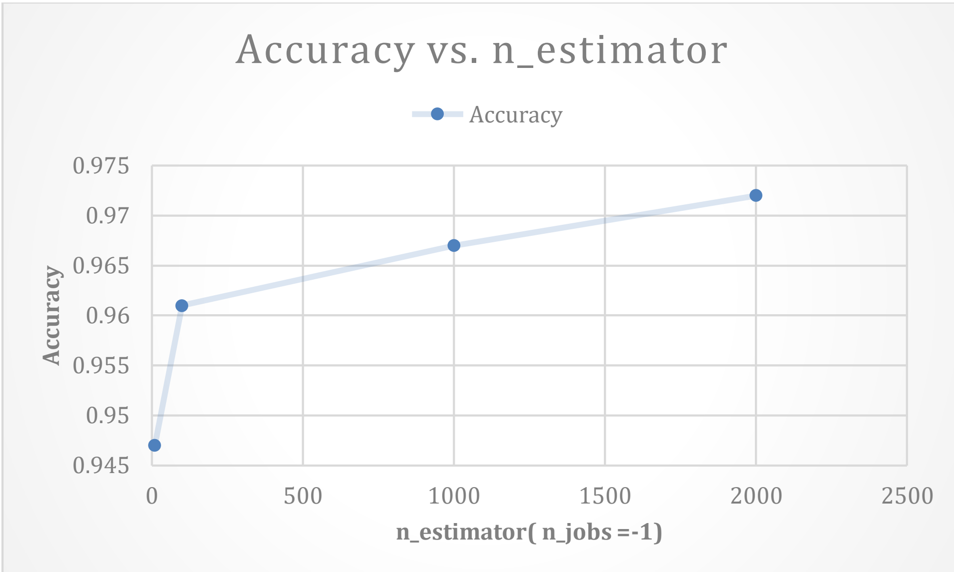
3 Random Forest

The random forest classifier was run on the MNIST dataset with different hyperparameters. It was seen the best performance came from not restricting the max_leaf_nodes and using a high number of n_estimator like 1000. The n_jobs parameter was set to -1. The results that were obtained are given below.

```
Random forest accuracy on MNIST data:  0.9708
random forest accuracy on USPS data:  0.112455622781
```

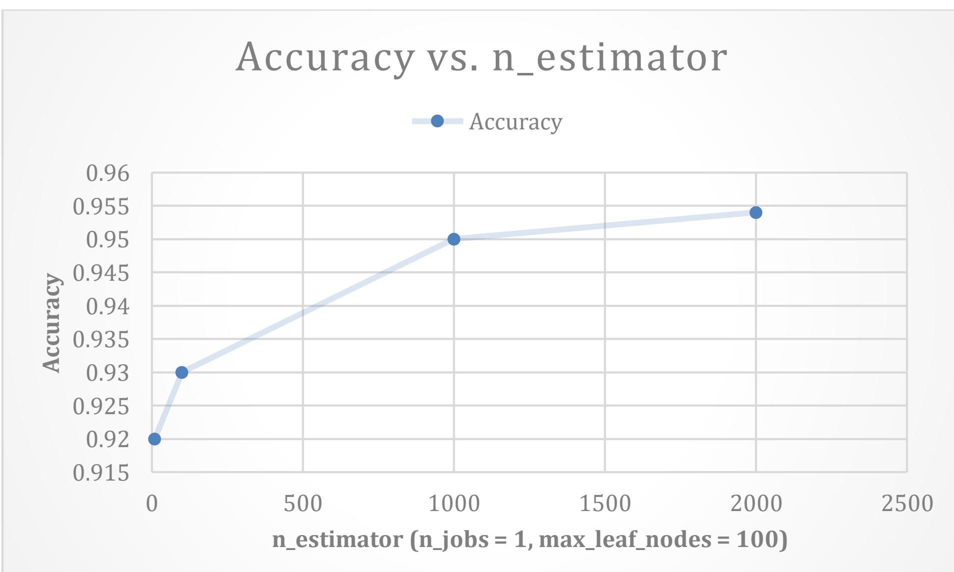
3.1 Number of Trees in Forest (n_estimators)

It was seen that as the number of trees increased the accuracy of the model also increased.



3.2 Number of Max_Leaf_Nodes and n_jobs

It was seen that if max_leaf_nodes are decreased and n_jobs is set at value 1, then the accuracy decreases. It is reasonable as the trees cannot flourish to their maximum potential.



4 Support Vector Machine

LinearSVC was used to calculate the accuracy at first setting every other parameter at the default value. The results obtained are given below.

```
Linear SVM accuracy for MNIST data:  0.8538
Linear SVM accuracy for USPS data:  0.123756187809
```

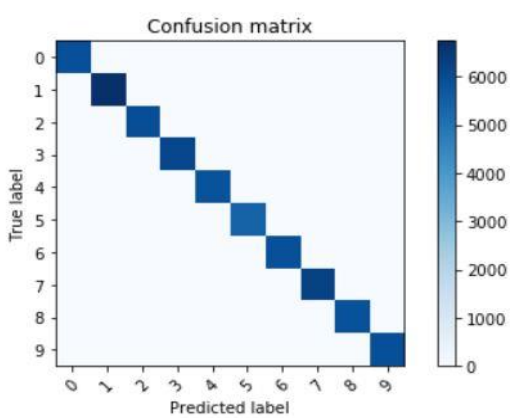
4.1 SVM with different kernels

Using the default value for kernel produced an overfitting model that could not predict the digits of the test set accurately. When SVC with linear kernel was used it produced a good model. The accuracy was seen to be around 86% for MNIST data. The reason for this is that empty spaces between different classes are created with a Gaussian function when using radial basis functions and in case of linear kernel, the model is less tunable and is basically a linear interpolation. Moreover, the nonlinear function has a higher variance and linear kernel has a lower variance.

4.2 SVM with radial basis functions having different gamma values

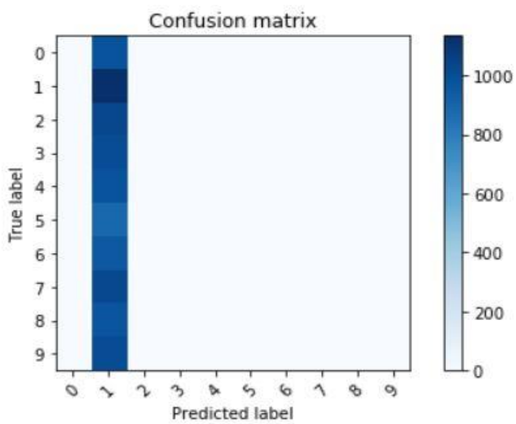
When the gamma value was set to 1, then the model became hugely overfitted with a training test accuracy of 100%. As it was not a general model it performed poorly on the test datasets of MNIST and USPS. The model showed an accuracy of around 11% on the test set of data. A better illustration of the model would be through a confusion matrix which is shown below.

```
Confusion Matrix of MNIST Train data(Gamma = 1)

Confusion matrix, without normalization
[[5923    0    0    0    0    0    0    0    0    0]
 [   0 6742    0    0    0    0    0    0    0    0]
 [   0    0 5958    0    0    0    0    0    0    0]
 [   0    0    0 6131    0    0    0    0    0    0]
 [   0    0    0    0 5842    0    0    0    0    0]
 [   0    0    0    0    0 5421    0    0    0    0]
 [   0    0    0    0    0    0 5918    0    0    0]
 [   0    0    0    0    0    0    0 6265    0    0]
 [   0    0    0    0    0    0    0    0 5851    0]
 [   0    0    0    0    0    0    0    0    0 5949]]
```

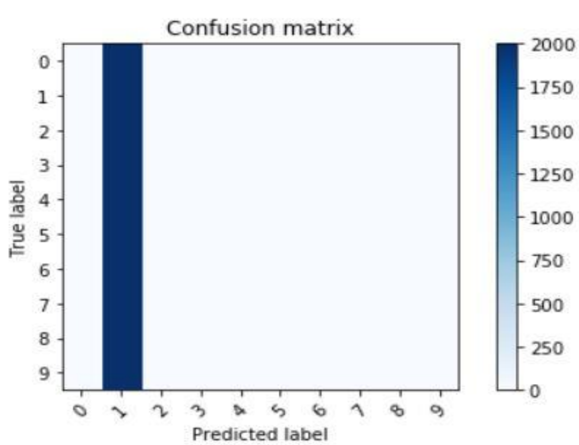


It can be seen that the model predicted everything correctly in the training set when gamma =1. It shows the model is hugely overfitted. The prediction of the model can be seen in the later confusion matrices which shows that the model almost always predicted every digit as 1 both in the MNIST and the USPS dataset.

```
Confusion Matrix of MNIST Test data(Gamma = 1)

Confusion matrix, without normalization
[[   0  980    0    0    0    0    0    0    0    0]
 [   0 1135    0    0    0    0    0    0    0    0]
 [   0 1032    0    0    0    0    0    0    0    0]
 [   0 1010    0    0    0    0    0    0    0    0]
 [   0  982    0    0    0    0    0    0    0    0]
 [   0  892    0    0    0    0    0    0    0    0]
 [   0  958    0    0    0    0    0    0    0    0]
 [   0 1028    0    0    0    0    0    0    0    0]
 [   0  974    0    0    0    0    0    0    0    0]
 [   0 1009    0    0    0    0    0    0    0    0]]
```




```
Confusion Matrix of USPS Test data(Gamma = 1)

Confusion matrix, without normalization
[[   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 1999    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]
 [   0 2000    0    0    0    0    0    0    0    0]]
```



The non-linear radial basis function has a high variance and that could be decreased using regularization. Using the C parameter in SVM the accuracy was seen to have increased in case of the radial basis function kernel.

5 Answer to Questions

Q.1 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?

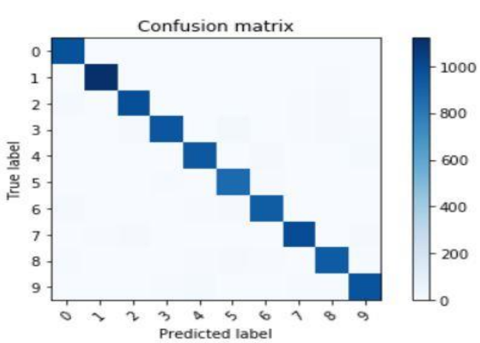
The **No Free Lunch** theorem says that there is no superior black-box optimization strategy. That is, we cannot device a model in such a way that it can give optimal results in all test data sets. In our case, the results support the theorem. This is because even though MNIST and USPS datasets have the same numerical digits (0-9), after training our model on MNIST dataset, when we tested the model against the USPS dataset the accuracy in prediction was really poor. Whereas we could obtain around 90% accuracy in all of our models with MNIST dataset, the accuracy in prediction for the USPS dataset was only around 20-30 %. These results go on to show that training a machine learning model in one set of data cannot give the same performance in another set of data.

Q.2 Observe the confusion matrix of each classifier and describe the relative strengths/weaknesses of each classifier. Which classifier has the overall best performance?

While passing the training and testing sets in the model first to_categorical of keras.utils was used to convert class vectors to binary class matrices. After training of the model the predictions were converted back to digits from binary class matrices using a decode function so that we can create a confusion matrix.

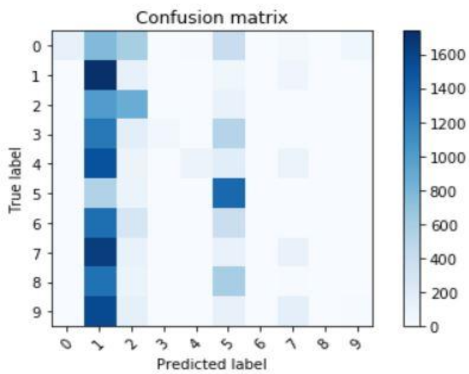
Neural Network Confusion Matrix (MNIST dataset)

```
Confusion matrix, without normalization
[[ 972    0    1    2    0    1    2    1    1    0]
 [    0 1120    2    1    0    1    3    1    7    0]
 [   11    4  987    4    1    3    4    5   12    1]
 [    2    0    9  954    0   23    0    6   11    5]
 [    1    0    4    0  947    0    9    1    4   16]
 [    4    1    0    8    2  863    6    1    4    3]
 [   12    3    0    0    5    9  926    0    3    0]
 [    1    7   10    2    1    0    0  995    2   10]
 [    6    2    1    4    6   10    5    4  930    6]
 [    3    4    1    5   15    3    1    5    6  966]]
```



Neural Network Confusion Matrix (USPS dataset)

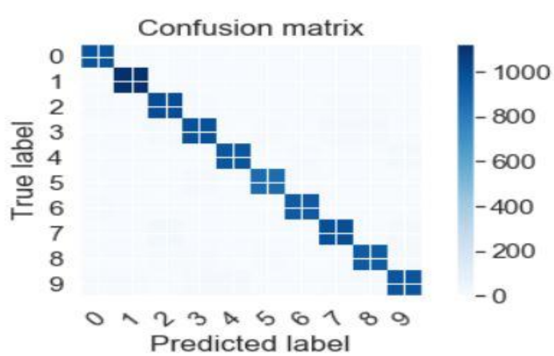
```
Confusion matrix, without normalization
[[ 125  769  591    0   12  400    0   40    0   63]
 [    0 1737  131    0    0   62    0   70    0    0]
 [    0 1000  879    0    0  119    0    1    0    0]
 [    0 1251  172   46    0  525    0    6    0    0]
 [    2 1518   97    0   98  189    0   91    0    5]
 [    0  546   91    0    1 1359    0    3    0    0]
 [    4 1322  276    0    3  390    1    4    0    0]
 [    0 1645  117    0    0  117    0  121    0    0]
 [    1 1300   90    0    0  598    0   11    0    0]
 [    0 1555  146    2    1  129    0  157    0   10]]
```



From the confusion matrix in my neural network model it can be seen that the model could almost accurately predict the digits in case of MNIST dataset, but when trying to predict the digits in USPS dataset the model had a tendency to predict all of the digits as the digit one and digit 5 in some cases. The neural network model was really successful in predicting the digit 1 in the MNIST dataset.

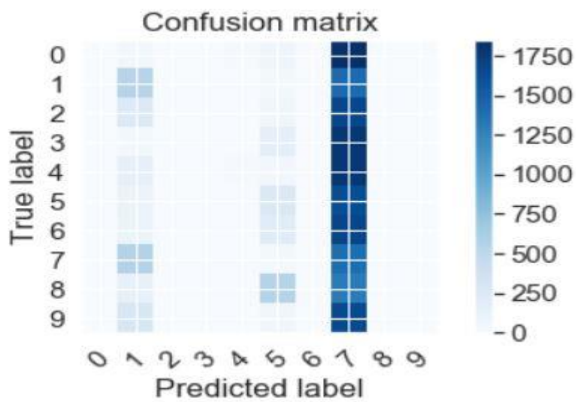
Random Forest Confusion Matrix (MNIST Dataset)

```
Confusion matrix, without normalization
[[ 970    0    0    0    0    2    2    1    4    1]
 [    0 1124    2    3    0    2    2    0    1    1]
 [    6    0  999    6    3    0    4    8    6    0]
 [    0    0    9  976    0    5    0    9    9    2]
 [    1    0    1    0  958    0    5    0    2   15]
 [    3    0    0   11    3  862    6    2    4    1]
 [    7    3    0    0    2    3  940    0    3    0]
 [    1    2   18    1    1    0    0  992    1   12]
 [    4    0    6    7    2    5    3    4  934    9]
 [    5    5    2    9   10    3    1    5    5  964]]
```



Random Forest Confusion Matrix (USPS Dataset)

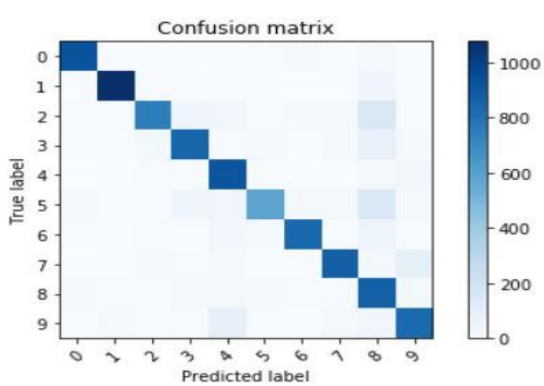
```
Confusion matrix, without normalization
[[   0   53    0    0   18   83    0 1846    0    0]
 [   0  544    0    0    0   29    0 1427    0    0]
 [   0  248    1    0    1   71    0 1678    0    0]
 [   0   35    0    0    6  166    0 1793    0    0]
 [   0  146    0    0   15   44    0 1795    0    0]
 [   0  104    0    0    5  259    0 1632    0    0]
 [   0   98    0    0    1  210    0 1691    0    0]
 [   0  564    0    0    0   33    0 1403    0    0]
 [   0  136    0    0    6  550    0 1308    0    0]
 [   0  283    0    0    5   59    0 1653    0    0]]
```



From the confusion matrix in my random forest model it can be seen that the model could almost accurately predict the digits in case of MNIST dataset, but when trying to predict the digits in USPS dataset the model had a tendency to predict all of the digits as the digit seven. The random forest model was really successful in predicting the digit 1 in the MNIST dataset.

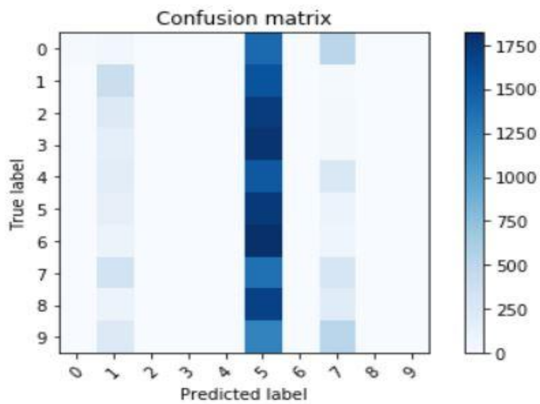
Support Vector Machine Confusion Matrix (MNIST Dataset)

```
Confusion matrix, without normalization
[[ 928    0    3    3    5    8   11    4   16    2]
 [   0 1077    2    2    0    0    4    2   47    1]
 [   6    7  749   47   37    4   13   13  154    2]
 [   6    2   18  853    8   13    4   13   84    9]
 [   1    1    6    0  921    0    3    2   18   30]
 [  12    4    3   50   35  581   18   18  150   21]
 [   8    3    6    1   27   15  843    1   54    0]
 [   0    4    9    6   19    0    1  877   21   91]
 [  10    6    4   20   21   12    6    5  872   18]
 [   3   12    1    7   86    2    0   22   39  837]]
```



Support Vector Machine Confusion Matrix (USPS Dataset)

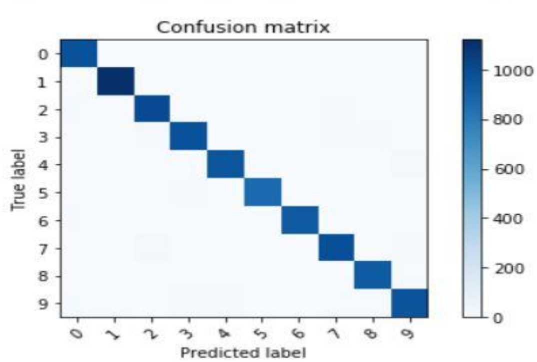
```
Confusion matrix, without normalization
[[  20   51    0    0    0 1414    0  515    0    0]
 [   0  404    0    0    0 1570    0   26    0    0]
 [   5  228    0    0    0 1730    0   36    0    0]
 [   5  167    0    0    0 1787    0   41    0    0]
 [   3  189    0    0    6 1534    0  268    0    0]
 [   3  144    0    0    0 1755    0   98    0    0]
 [   3   94    0    0    1 1820    3   79    0    0]
 [   3  340    0    0    0 1370    0  287    0    0]
 [   0   96    0    0    1 1694    0  209    0    0]
 [   4  240    0    0    2 1233    0  521    0    0]]
```



From the confusion matrix in my support vector machine model it can be seen that the model could almost accurately predict the digits in case of MNIST dataset, but when trying to predict the digits in USPS dataset the model had a tendency to predict all of the digits as the digit five. The neural network model was really successful in predicting the digit 1 in the MNIST dataset however in comparison to neural network and random it performed relatively worse in predicting all of the other digits.

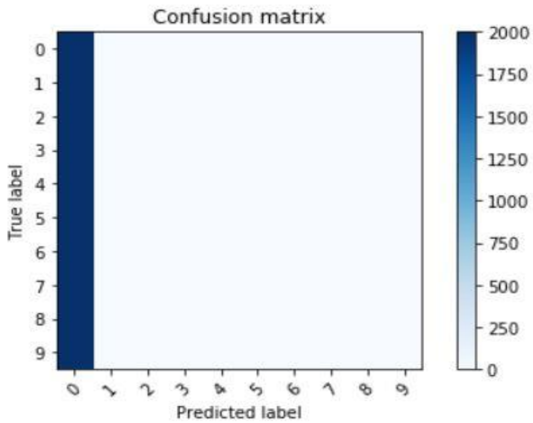
Logistic Regression Confusion Matrix (MNIST Dataset)

```
Confusion matrix, without normalization
[[ 971    0    0    0    0    3    3    1    2    0]
 [   0 1120    3    3    0    2    3    1    3    0]
 [   6    0  999    3    3    1    4   10    6    0]
 [   0    0   10  971    0    8    0    9    9    3]
 [   1    0    2    0  954    0    4    1    2   18]
 [   3    0    0   13    3  862    4    1    4    2]
 [   6    3    1    0    3    4  939    0    2    0]
 [   1    3   23    2    0    0    0  986    3   10]
 [   4    0    7    5    5    4    3    4  934    8]
 [   7    5    2   12    9    1    1    5    6  961]]
```



Logistic Regression Confusion Matrix (USPS Dataset)

```
Confusion matrix, without normalization
[[2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [1999    0    0    0    0    0    0    0    0    0]
 [1999    1    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]
 [2000    0    0    0    0    0    0    0    0    0]]
```



From the confusion matrix in my logistic regression model it can be seen that the model could almost accurately predict the digits in case of MNIST dataset, but when trying to predict the digits in USPS dataset the model had a tendency to predict all of the digits as the digit zero. Similar to the other models that I have designed, logistic regression model was really successful in predicting the digit 1 in the MNIST dataset.

From the analysis of the different models it can be seen that the best performance was achieved by the neural network and random forest model. However, these models excelled in predicting different digits accurately. The results are summarized as follows.

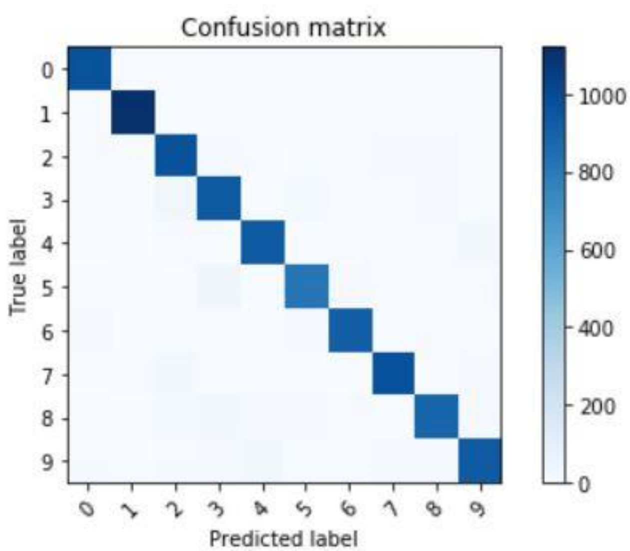
Digit 0: Neural Network
Digit 1: Random Forest
Digit 2: Random Forest
Digit 3: Random Forest
Digit 4: Random Forest
Digit 5: Neural Network
Digit 6: Random Forest
Digit 7: Neural Network
Digit 8: Random Forest
Digit 9: Neural Network

Q.3 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?

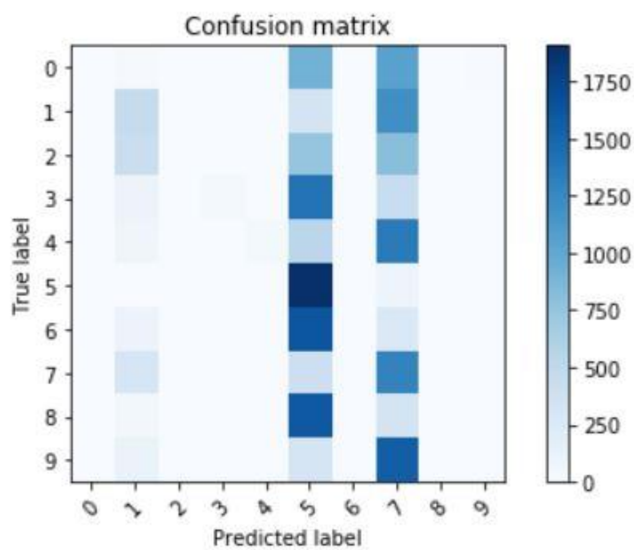
The ensemble classifier model that I have used is the majority voting classifier. It was a combination of the four models: Multilayer Perceptron Neural Network, Logistic Regression, Random Forest and Support Vector Machine. Through the combination of the four models it was seen that the combined performance showed a better performance than support vector machine and logistic regression but performed a little worse than neural network and random forest. This is with respect to the best model that I could make from neural network and random forest. If I had used less n_estimators in random forest or less nodes and dense layers in multilayer perceptron neural network the majority voting classifier would be at the top. However, it was seen that majority voting performed really worse than the individual models when trying to predict the USPS dataset. To understand the effect of the models on the classification accuracy I decided to make a combined model using Logistic Regression, Random Forest and Neural Network. It was seen that the accuracy of the digit prediction was around 92%. This showed the neural network contributed a lot in determining a higher accuracy of the entire model.

The confusion matrices of the majority voting classifier for the MNIST and USPS datasets are given below.

```
Confusion matrix, without normalization
[[ 968    1    1    0    0    3    4    1    2    0]
 [   0 1120    4    4    0    2    3    0    2    0]
 [   8    2  970    9    7    4    5   13   11    3]
 [   5    2   27  936    1   16    0    8   12    3]
 [   3    2    8    1  935    1    6    2    5   19]
 [   7    3    4   36    4  814    9    1    7    7]
 [  12    3    3    2    4   10  918    0    6    0]
 [   3    5   22    2    3    1    0  976    4   12]
 [   6    0   17   21    9    9    6    3  891   12]
 [   7    4    5   13   23    4    1    9   13  930]]
```



```
Confusion matrix, without normalization
[[   2   19    1    0    5  914    0 1051    0    8]
 [   0  466    0    0    0  333    0 1201    0    0]
 [   0  428    7    0    0  755    0  809    0    0]
 [   0  107    0   43    0 1408    0  442    0    0]
 [   0   77    0    0   43  532    0 1347    0    1]
 [   0    5    0    0    0 1905    0   90    0    0]
 [   0  106    2    0    1 1625    0  266    0    0]
 [   0  300    0    0    0  406    0 1294    0    0]
 [   0   58    0    0    0 1606    0  336    0    0]
 [   0  122    0    0    1  317    0 1558    0    2]]
```



From the confusion matrix it can be seen that the ensemble classifier predicted almost accurately the MNIST dataset but in case of USPS dataset it predicted it to be 5 and 7 in most cases.

The accuracy of Majority Voting Classifier in MNIST Dataset: 95%
The accuracy of Majority Voting Classifier in USPS Dataset: 31%