# Project 3

## Classification

Illa Nuka Saranya
50248926

21-Nov-18
Saranya@buffalo.edu

# 1. PROBLEM DEFINITION

The goal of this project is to implement machine learning methods using Logistic Regression, Neural Network, Random Forest and Support Vector Machine (SVM) for the task of classification of recognizing a 28×28 grayscale handwritten digit image and identify it as a digit among 0, 1, 2, ... , 9. First, we implement an ensemble of four classifiers for a given task and the results of all the individual classifiers are the combined to make a final decision.

**DATASET:**

- Both training and testing are performed on MNIST (Modified National Institute of Standards and Technology database) data which is a large database of handwritten digits contains 60,000 training images and 10,000 testing images.
- Also, we perform testing on USPS Data to check if the above trained model is generalized and this dataset consists of 2000 images for every digit

**TASKS**:

- To find if the testing results support “No Free Lunch” theorem.
- To describe the relative strengths/weaknesses of each classifier by observing its confusion matrix to determine the classifier with overall best performance.
- To combine the results of the individual classifiers using a classifier combination method such as majority voting and check if the combined performance is better than that of any individual classifier.

# 2. IMPLEMENTATION

**Confusion Matrix**: While implementing various types of classifiers, we compute **Confusion matrix** which is a matrix (table) that can be used to measure the performance of a machine learning algorithm, usually a supervised learning one. Each row of the confusion matrix represents the instances of an actual class and each column represents the instances of a predicted class.

- Confusion matrix is helpful in measuring accuracy under two circumstances where the data has uneven number of classes in training or when there are more than two classes.
- The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix.
- **The confusion matrix shows the ways in which the classification model is confused when it makes predictions.**

- It gives us insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.It is this breakdown that overcomes the limitation of using classification accuracy alone.

## 2.1 Logistic Regression

### 2.1.1 Extract feature values and labels from the data:

- Processing the MNIST dataset into a Numpy array that contains the feature vectors (input vectors) and a Numpy array that contains the labels (target vectors).
- As the downloaded dataset from internet is a compressed pickle file which is dumped using python pickle module, we unpickle the data which internally contains the fixed sets of training, validation and testing.
- Here, each dataset is in the form of a tuple of numpy arrays where the first array represents the feature vectors and the other one contains the labels.

### 2.1.2 Data Partition

The MNIST dataset is originally partitioned into a training set, validation and testing set.

| Tuple | Feature Vectors | Class labels |
|---|---|---|
| training_data | training_data[0]<br><br>(50000,784) | training_data[0]<br><br>(50000,) |
| validation_data | validation_data[0]<br><br>(10000,784) | validation_data[1]<br><br>(10000,) |
| testing_data | testing_data[0]<br><br>(10000,784) | testing_data[1]<br><br>(10000,) |

### 2.1.3 Train model parameter

- For a given group of hyper-parameters such as the number of iterations and the learning rate, train the model parameters on the training set.
- To generate probabilities, logistic regression uses a function that gives outputs between 0 and 1 for all values of X. We use sigmoid function here.
- In order to fit the model for 10 classes, we convert the training target values by using one hot encoding. For that we can also use to_categorical method that is there in keras. Thus, by using sigmoid functions, we will be able to extend this classification on 10 classes as well.

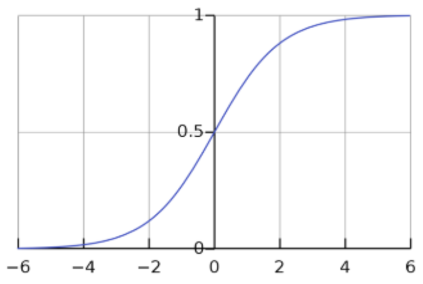


Fig. 2—Logistic function

$$h_\theta(x) = g(\theta^T x)$$

$$z = \theta^T x$$

$$g(z) = \frac{1}{1 + e^{-z}}$$

**Loss Function**:

$$h = g(X\theta)$$

$$J(\theta) = \frac{1}{m} \cdot \left(-y^T \log(h) - (1-y)^T \log(1-h)\right)$$

**Gradient Descent**: Goal is to minimize the loss function. Finding partial derivate of loss function with respect to each weight, we have

$$\frac{\delta J(\theta)}{\delta \theta_j} = \frac{1}{m} X^T (g(X\theta) - y)$$

### 2.1.4 Tune hyper-parameters

Validate the classification performance of model on the validation set. Tuning the hyper-parameters, by changing the number of iterations and learning rate, the following values of the accuracy are tabulated.

| Learning Rate | Iteration | Training Accuracy | Validation Accuracy |
|---|---|---|---|
| **0.01** | 100 | 0.74 | 0.73 |
| **0.001** | 100 | 0.74 | 0.74 |
| **0.001** | 500 | 0.841 | 0.852 |
| **0.001** | 1000 | 0.653 | 0.663 |
| **0.0001** | 500 | 0.856 | 0.864 |
| **0.0001** | 1000 | 0.842 | 0.854 |

From the above observations, we choose the model with learning rate= and number of iterations= as the best one as it gives the best accuracy and doesn't over fit.

### 2.1.5 Evaluate on testing sets

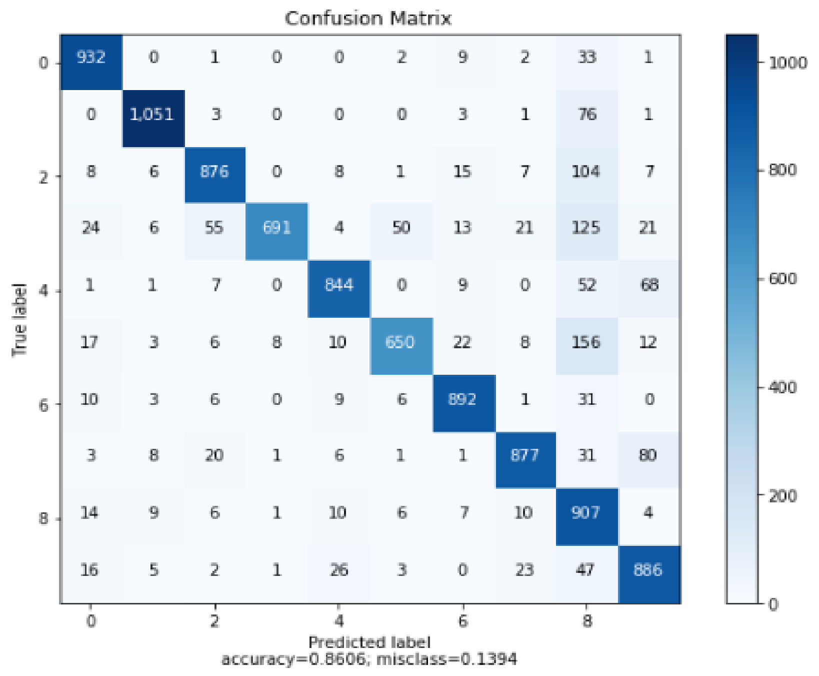
- Test the trained models on both MNIST test set and USPS data.
- We test the best-fit model obtained on MNIST test, on the USPS dataset. The final best accuracy obtained on both MNIST and USPS dataset are as below.

| Learning Rate | Iteration | MNIST Testset | USPS Testset |
|---|---|---|---|
| 0.01 | 100 | 0.725 | 02813 |
| 0.001 | 100 | 0.73 | 0.2865 |
| 0.001 | 500 | 0.847 | 0.314 |
| 0.001 | 1000 | 0.658 | 0.2534 |
| **0.0001** | **500** | **0.861** | **0.3201** |
| 0.0001 | 1000 | 0.847 | 0.8413 |

### 2.1.6 Confusion Matrix:

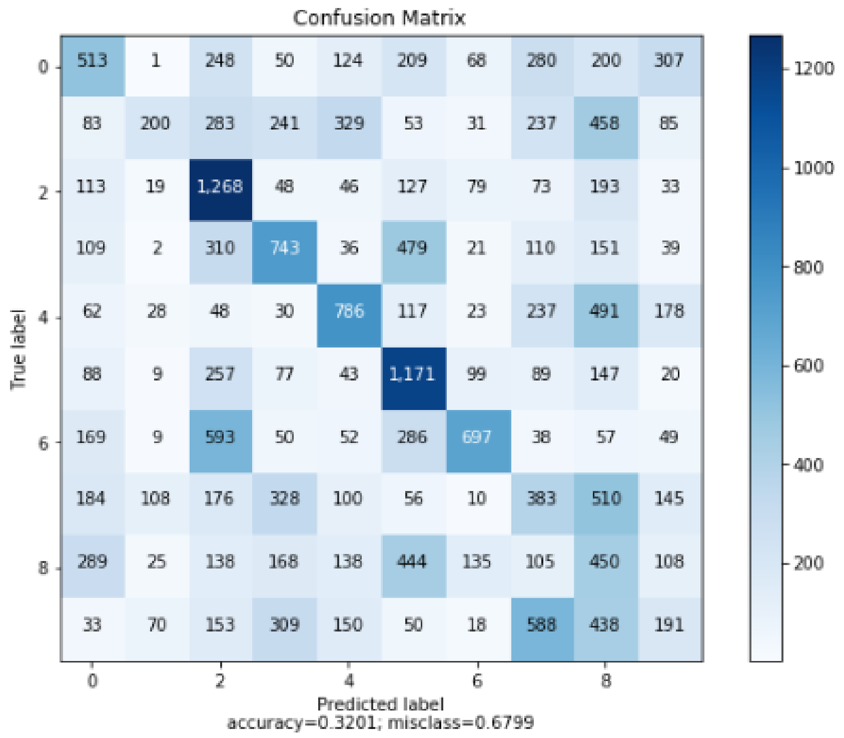
When the learning rate = 0.0001 and number of iterations = 500, we get the best accuracy for both MNIST and USPS dataset. The following are the confusion matrix for those hyper-parameters.

Confusion Matrix on MNIST dataset



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 932 | 0 | 1 | 0 | 0 | 2 | 9 | 2 | 33 | 1 |
| 0 | 1,051 | 3 | 0 | 0 | 0 | 3 | 1 | 76 | 1 |
| 8 | 6 | 876 | 0 | 8 | 1 | 15 | 7 | 104 | 7 |
| 24 | 6 | 55 | 691 | 4 | 50 | 13 | 21 | 125 | 21 |
| 1 | 1 | 7 | 0 | 844 | 0 | 9 | 0 | 52 | 68 |
| 17 | 3 | 6 | 8 | 10 | 650 | 22 | 8 | 156 | 12 |
| 10 | 3 | 6 | 0 | 9 | 6 | 892 | 1 | 31 | 0 |
| 3 | 8 | 20 | 1 | 6 | 1 | 1 | 877 | 31 | 80 |
| 14 | 9 | 6 | 1 | 10 | 6 | 7 | 10 | 907 | 4 |
| 16 | 5 | 2 | 1 | 26 | 3 | 0 | 23 | 47 | 886 |

## Confusion matrix on USPS dataset



The following one is the classification report of the metrics such as precision,recall and f1-score for USPS dataset.

```
              precision    recall  f1-score   support

           0       0.23      0.36      0.28      1245
           1       0.08      0.46      0.14       357
           2       0.72      0.32      0.44      4549
           3       0.54      0.31      0.39      3450
           4       0.32      0.53      0.40      1206
           5       0.55      0.41      0.47      2694
           6       0.27      0.61      0.37       870
           7       0.15      0.17      0.16      1725
           8       0.21      0.15      0.17      2897
           9       0.08      0.17      0.11      1006

   micro avg       0.31      0.31      0.31     19999
   macro avg       0.31      0.35      0.29     19999
weighted avg       0.42      0.31      0.34     19999
```

From the above 2, it is observed that, the logistic regression algorithm finds it difficult to predict the values of 9,1 and 7 in the order. It is also observed that the model has incorrectly predicted 8 for more number of times instead of other actual values values. Number 2 has been predicted the best in logistic regression model.

The above results support the "**No Free Lunch**" theorem which states that though the model has obtained an accuracy of 86% on MNIST testing dataset, it could not perform well on USPS dataset which only gave 32% accuracy when tested.

## 2.2 Neural Network

### 2.2.1 Extract feature values and labels from the data:

- Processing the MNIST dataset is done in the same way as logistic regression.

### 2.2.2 Data Partition

- The MNIST dataset is originally partitioned into a training set, validation and testing set as in the logistic regression.

### 2.2.3 Train model parameter

For a given group of hyper-parameters such as the number of hidden-layers, hidden nodes learning rate, we train the model parameters on the training set.
The following values of accuracy are tabulated, for set of different parameters.

- We take input_size = 784 as there are 784 features.
- The second dense layer nodes = 10 as there are 10 classes or labels.
- Out of all the nodes, to avoid over-fitting, we do drop-out few neurons. Here we drop out 20% of the total nodes.

Using Adam optimizer(one of the best optimizers) and categorical_crossentropy as loss function(as there are more than two classes), we change the following parameters to find the one that fits the best.

### 2.2.4 Tuning Hyper Parameters

We tune the model by changing the following values and tabulate validation accuracy to find the best fit among the below ones.

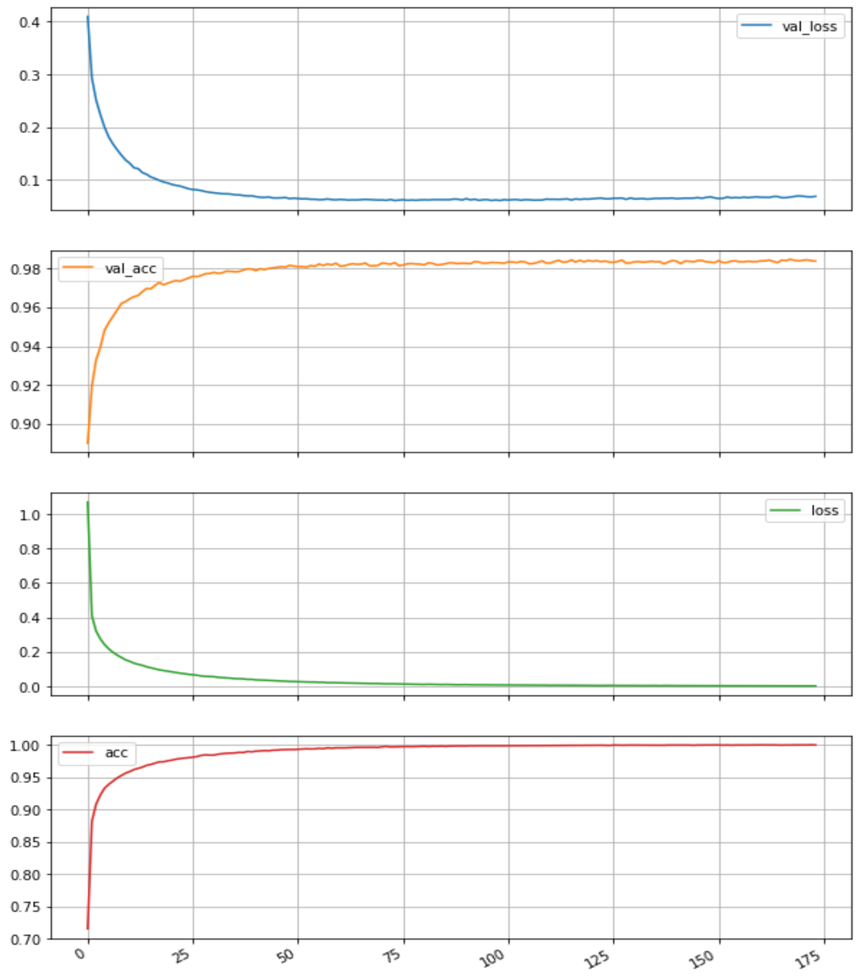
| Parameters | Values | | | | |
|---|---|---|---|---|---|
| No of hidden layers | 1 | 1 | 2 | 2 | 3 |
| No of nodes | 128 | 256 | 128 | 256 | 128 |
| epochs | 200 | 300 | 200 | 300 | 200 |
| Model_batch_size | 1024 | 2048 | 1024 | 2048 | 1024 |
| Tb_batch_size | 32 | 64 | 32 | 64 | 32 |
| Validation Accuracy | 0.9791 | 0.9821 | 0.9813 | 0.9833 | 0.983 |

### 2.2.5 Evaluate on testing sets:

Test the trained models on both MNIST test set and USPS data. We test the best-fit model obtained on MNIST test, on the USPS dataset. The final best accuracy obtained on both MNIST and USPS dataset are as below.

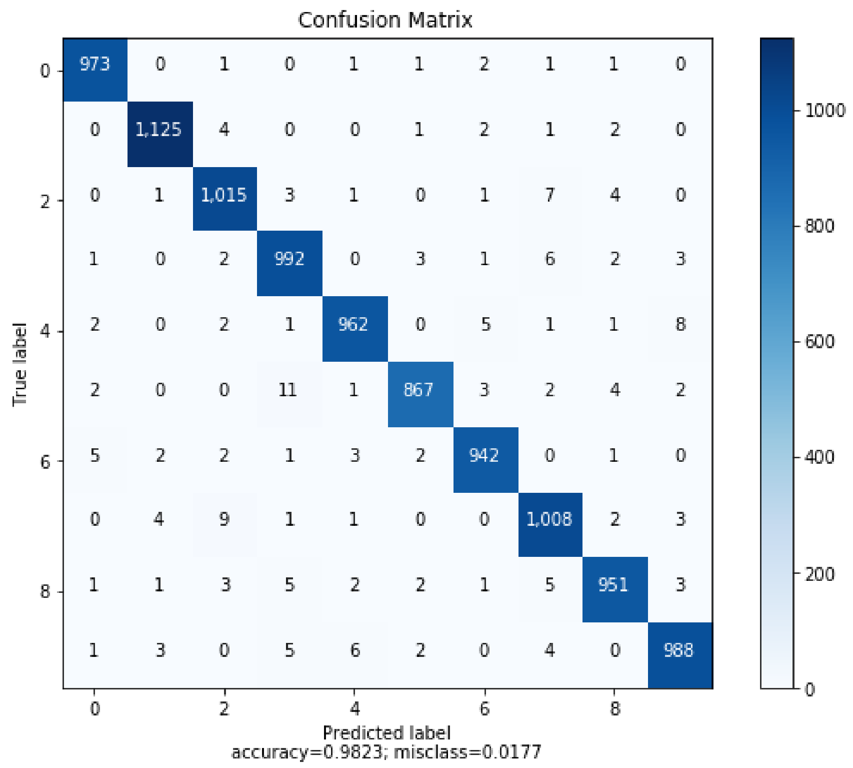
| Parameters | Values | | | | |
|---|---|---|---|---|---|
| No of hidden layers | 1 | **1** | 2 | 2 | 3 |
| No of nodes | 128 | **256** | 128 | 256 | 128 |
| epochs | 200 | **300** | 200 | 300 | 200 |
| Model_batch_size | 1024 | **2048** | 1024 | 2048 | 1024 |
| Tb_batch_size | 32 | **64** | 32 | 64 | 32 |
| MNIST Testing | 0.9794 | **0.9822** | 0.9803 | 0.9822 | 0.9794 |
| USPS Testing | 0.4058 | 0.4287 | 0.4335 | 0.4504 | 0.4511 |
| USPS Loss | 6.4224 | **5.4334** | 6.4184 | 6.0094 | 6.2160 |

The following graph is observed for no of epochs and accuracy of validation and testing sets of MNIST dataset.

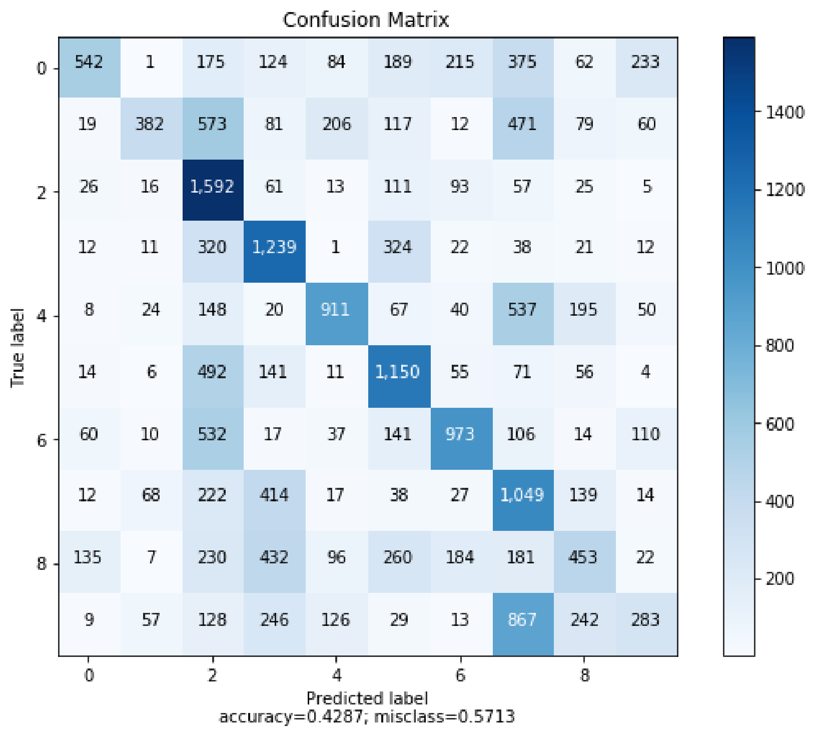


### 2.2.6 Confusion Matrix:

Confusion matrix on MNIST testing dataset



## Confusion matrix of USPS dataset



| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 542 | 1 | 175 | 124 | 84 | 189 | 215 | 375 | 62 | 233 |
| 19 | 382 | 573 | 81 | 206 | 117 | 12 | 471 | 79 | 60 |
| 26 | 16 | 1,592 | 61 | 13 | 111 | 93 | 57 | 25 | 5 |
| 12 | 11 | 320 | 1,239 | 1 | 324 | 22 | 38 | 21 | 12 |
| 8 | 24 | 148 | 20 | 911 | 67 | 40 | 537 | 195 | 50 |
| 14 | 6 | 492 | 141 | 11 | 1,150 | 55 | 71 | 56 | 4 |
| 60 | 10 | 532 | 17 | 37 | 141 | 973 | 106 | 14 | 110 |
| 12 | 68 | 222 | 414 | 17 | 38 | 27 | 1,049 | 139 | 14 |
| 135 | 7 | 230 | 432 | 96 | 260 | 184 | 181 | 453 | 22 |
| 9 | 57 | 128 | 246 | 126 | 29 | 13 | 867 | 242 | 283 |

It is observed from the confusion matrix and the classification report that the neural network finds it difficulty in correctly predicting the values of 9,1, 8 and 0

```
              precision    recall  f1-score   support

           0       0.23      0.58      0.32       781
           1       0.17      0.65      0.27       521
           2       0.79      0.39      0.52      4073
           3       0.61      0.42      0.50      2873
           4       0.49      0.57      0.53      1720
           5       0.63      0.46      0.53      2732
           6       0.49      0.54      0.52      1800
           7       0.49      0.26      0.34      3768
           8       0.20      0.39      0.27      1053
           9       0.11      0.33      0.17       678

   micro avg       0.42      0.42      0.42     19999
   macro avg       0.42      0.46      0.40     19999
weighted avg       0.54      0.42      0.45     19999
```

The above results support the “**No Free Lunch**” theorem which states that though the Deep Neural Network model has obtained an accuracy of **98%** on MNIST testing dataset, it could not perform well on USPS dataset which only gave **42%** accuracy when tested.

## 2.3 Random Forest

- Random forests, also known as random decision forests, are a popular ensemble method that can be used to build predictive models for both classification and regression problems. Ensemble methods use multiple learning models to gain better predictive results — in the case of a random forest, the model creates an entire forest of random uncorrelated decision trees to arrive at the best possible answer.
- Decision trees tend to have high variance when they utilize different training and test sets of the same data, since they tend to overfit on training data and thus leads to poor performance on unseen data which limits the usage of decision trees in predictive modeling.
- Using ensemble methods, we can create models that utilize underlying decision trees as a foundation for producing powerful results.

### 2.3.1 Extract feature values and labels from the data:

- Processing the MNIST dataset is done in the same way as logistic regression.

### 2.3.2 Data Partition

- The MNIST dataset is originally partitioned into a training set, validation and testing set as in the logistic regression.

### 2.3.3 Train model parameter

For a given group of hyper-parameters such as the number of estimators  and max_features, the following values of accuracy are tabulated, for set of different parameters.Rest of the parameters are taken as default.

```
{'bootstrap': True,
'class_weight': None,
'criterion': 'gini',
'max_depth': None,
'max_features': 'auto',
'max_leaf_nodes': None,
'min_impurity_decrease': 0.0,
'min_impurity_split': None,
'min_samples_leaf': 1,
'min_samples_split': 2,
'min_weight_fraction_leaf': 0.0,
'n_estimators': 'warn',
'n_jobs': None,
'oob_score': False,
'random_state': None,
'verbose': 0,
'warm_start': False}
```

### 2.3.4 Tuning Hyper Parameters

We tune the model by changing the following values and tabulate validation accuracy to find the best fit among the below ones. As there are quite a many possible combinations, we try changing only few hyper-parameters at random.

| Parameters | Values | | | |
|---|---|---|---|---|
| **No of estimators** | 10 | 20 | 100 | 200 |
| **No of max features** | sqrt | log2 | None | 50 |
| **USPS Testing** | 0.4058 | 0.4287 | 0.4335 | 0.4504 |

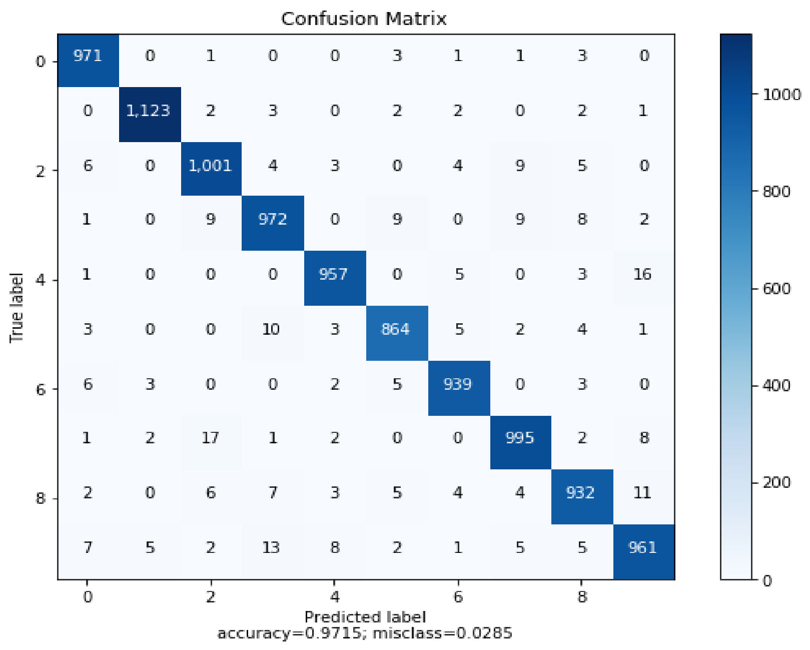
### 2.3.5 Evaluate on testing sets:

Test the trained models on both MNIST test set and USPS data. We test the best-fit model obtained on MNIST test, on the USPS dataset. The final best accuracy obtained on both MNIST and USPS dataset are as below.

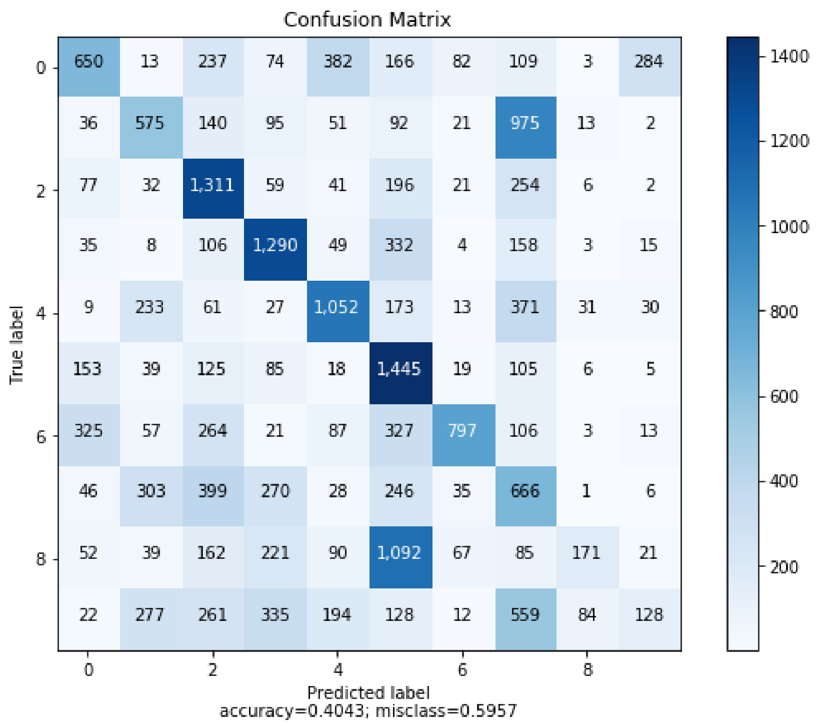
| Parameters | Values | | | |
|---|---|---|---|---|
| **No of estimators** | 10 | 20 | 100 | **200** |
| **No of max features** | sqrt | log2 | None | **50** |
| **USPS Testing** | 0.4058 | 0.4287 | 0.4335 | 0.4504 |
| **USPS Accuracy** | 0.3191 | 0.3267 | 0.3451 | **0.4042** |

### 2.3.6 Confusion Matrix:

Confusion matrix on MNIST dataset



## Confusion matrix on USPS dataset



The above results support the "**No Free Lunch**" theorem which states that though the Random Forest Algorithm model has obtained an accuracy of **97%** on MNIST testing dataset, it could not perform well on USPS dataset which only gave **40%** accuracy when tested.

```
              precision    recall  f1-score   support

           0       0.33      0.44      0.38      1491
           1       0.27      0.37      0.31      1465
           2       0.63      0.43      0.51      2913
           3       0.63      0.52      0.57      2417
           4       0.53      0.49      0.51      2139
           5       0.73      0.35      0.48      4154
           6       0.42      0.74      0.53      1132
           7       0.36      0.21      0.26      3465
           8       0.08      0.52      0.14       324
           9       0.07      0.27      0.11       499

   micro avg       0.40      0.40      0.40     19999
   macro avg       0.40      0.43      0.38     19999
weighted avg       0.51      0.40      0.43     19999
```

The random forest algorithm finds it difficult to correctly predict the values of 8 ,9 and 1. The numbers that were incorrectly predicted the most is 7 and 5. 8 is incorrectly predicted as 5 for 1092 times.

## 2.4 Support Vector Machine

- Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outliers detection.
- We use a kernel function to specify the decision function and uses only a subset of training points in the decision function called as support vectors.
- SVC, NuSVC and LinearSVC are classes capable of performing multi-class classification on a dataset.
- LinearSVC implements "one-vs-the-rest" multi-class strategy as SVC with linear kernel implements one-vs-one approach.

### 2.4.1 Extract feature values and labels from the data:

- Processing the MNIST dataset is done in the same way as logistic regression.

### 2.4.2 Data Partition

- The MNIST dataset is originally partitioned into a training set, validation and testing set as in the logistic regression.

### 2.4.3 Train model parameter

For a given group of hyper-parameters such as the type of the kernel the following values of accuracy are tabulated, for set of different parameters. Rest of the parameters are taken as default.

### 2.4.4 Tuning Hyper Parameters

We tune the model by changing the following values and tabulate validation accuracy to find the best fit among the below ones. As there are quite a many possible combinations, we try changing only few hyper-parameters at random.

| **Parameters** | **Values** | | | |
|---|---|---|---|---|
| **Kernel** | RBF | Linear | RBF | LinearSVC |
| **Gamma** | Default | Default | 1 | Default |
| **MNIST Validation** | 0.4058 | 0.4287 | 0.179 | 0.4335 |

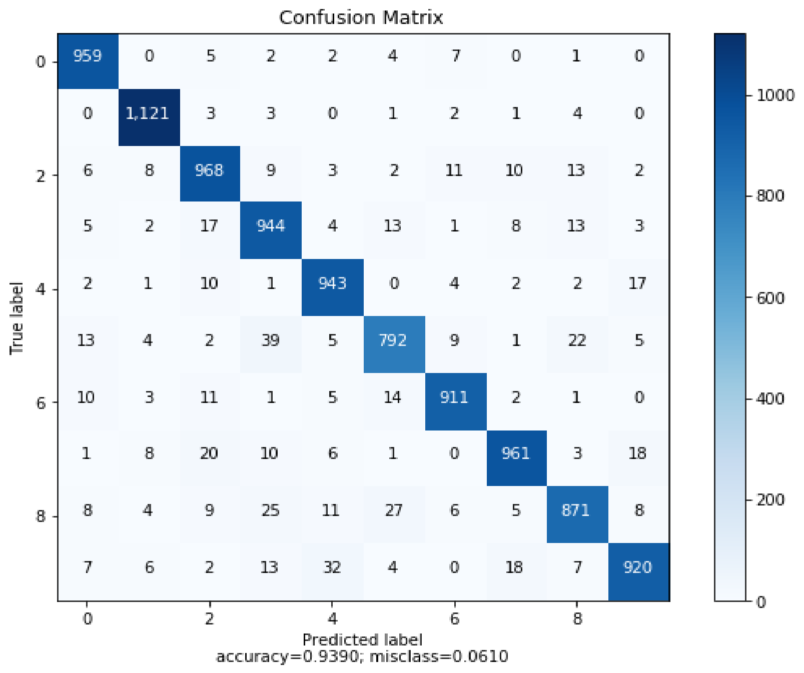
### 2.4.5 Evaluate on testing sets:

Test the trained models on both MNIST test set and USPS data. We test the best-fit model obtained on MNIST test, on the USPS dataset. The final best accuracy obtained on both MNIST and USPS dataset are as below.

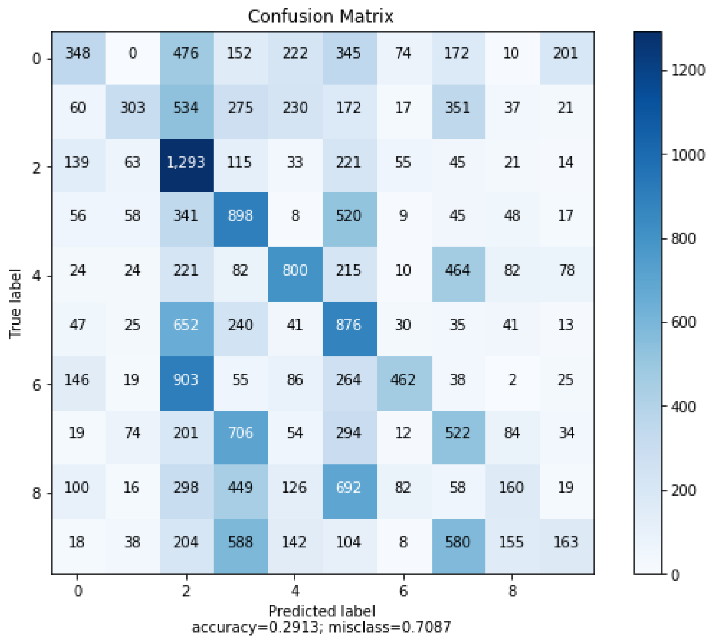
| Parameters | Values | | | |
|---|---|---|---|---|
| **Kernel** | RBF | Linear | RBF | Linear SVC |
| **Gamma** | Default | Default | 1 | Default |
| **MNIST Testing** | 0.4058 | 0.4287 | 0.18 | 0.4335 |
| **USPS Accuracy** | 0.3854 | 0.2912 | 0.09 | 0.2603 |

### 2.4.6 Confusion Matrix:

The above results support the "**No Free Lunch**" theorem which states that though the Support Vector Machine model has obtained an accuracy of **91%** on MNIST testing dataset, it could not perform well on USPS dataset which only gave **29%** accuracy when tested.



It is observed that the SVM predicts more incorrect values in its linear model. For most of the values observed, 2 and 5 are more predicted instead of other values. The predictions of actual 8,9 are very less.



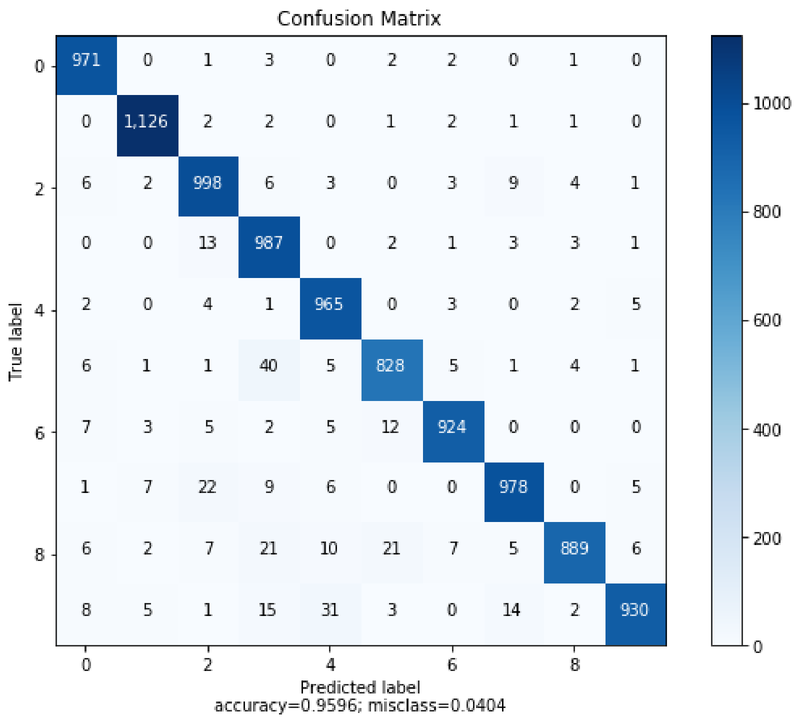
|  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|
| 348 | 0 | 476 | 152 | 222 | 345 | 74 | 172 | 10 | 201 |
| 60 | 303 | 534 | 275 | 230 | 172 | 17 | 351 | 37 | 21 |
| 139 | 63 | 1,293 | 115 | 33 | 221 | 55 | 45 | 21 | 14 |
| 56 | 58 | 341 | 898 | 8 | 520 | 9 | 45 | 48 | 17 |
| 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 |

accuracy=0.2913; misclass=0.7087

## 2.5 Combining classifier results

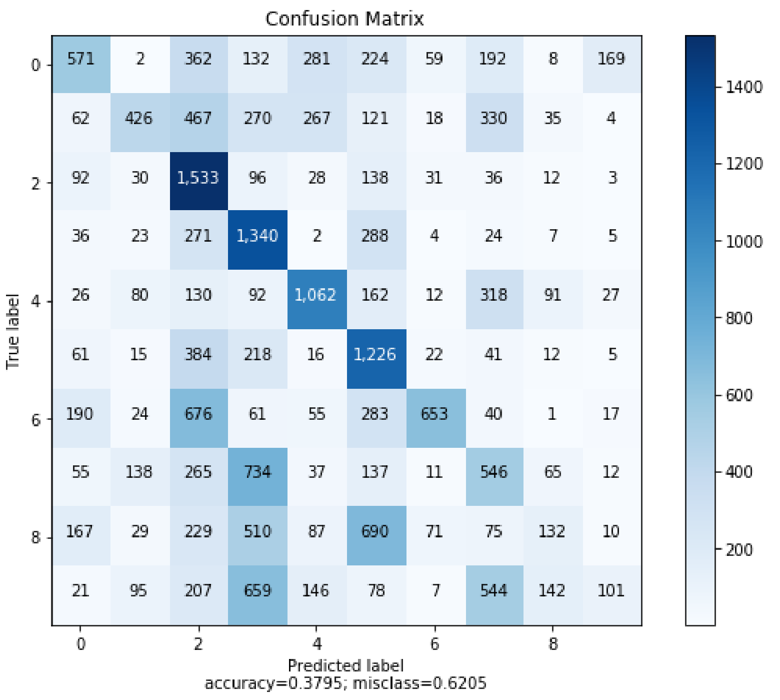
- Here we combined the results of 4 different classifiers using majority voting.
- We have 10000 target values, obtained for each of the 4 classifiers. We stack them in a way that all the target values appear in 4 columns.
- We then calculate the mode of the row, i.e the one that is predicted more by majority voting and form a new list of target values with these mode values of 10000 rows appended.
- The new list obtained is the combined classifier target values list and we test it for its accuracy.

Combined classifier accuracy obtained on MNIST dataset: 95.96



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 971 | 0 | 1 | 3 | 0 | 2 | 2 | 0 | 1 | 0 |
| 1 | 0 | 1,126 | 2 | 2 | 0 | 1 | 2 | 1 | 1 | 0 |
| 2 | 6 | 2 | 998 | 6 | 3 | 0 | 3 | 9 | 4 | 1 |
| 3 | 0 | 0 | 13 | 987 | 0 | 2 | 1 | 3 | 3 | 1 |
| 4 | 2 | 0 | 4 | 1 | 965 | 0 | 3 | 0 | 2 | 5 |
| 5 | 6 | 1 | 1 | 40 | 5 | 828 | 5 | 1 | 4 | 1 |
| 6 | 7 | 3 | 5 | 2 | 5 | 12 | 924 | 0 | 0 | 0 |
| 7 | 1 | 7 | 22 | 9 | 6 | 0 | 0 | 978 | 0 | 5 |
| 8 | 6 | 2 | 7 | 21 | 10 | 21 | 7 | 5 | 889 | 6 |
| 9 | 8 | 5 | 1 | 15 | 31 | 3 | 0 | 14 | 2 | 930 |

Combined Classifier accuracy obtained on USPS dataset: 37.9581



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 571 | 2 | 362 | 132 | 281 | 224 | 59 | 192 | 8 | 169 |
| 1 | 62 | 426 | 467 | 270 | 267 | 121 | 18 | 330 | 35 | 4 |
| 2 | 92 | 30 | 1,533 | 96 | 28 | 138 | 31 | 36 | 12 | 3 |
| 3 | 36 | 23 | 271 | 1,340 | 2 | 288 | 4 | 24 | 7 | 5 |
| 4 | 26 | 80 | 130 | 92 | 1,062 | 162 | 12 | 318 | 91 | 27 |
| 5 | 61 | 15 | 384 | 218 | 16 | 1,226 | 22 | 41 | 12 | 5 |
| 6 | 190 | 24 | 676 | 61 | 55 | 283 | 653 | 40 | 1 | 17 |
| 7 | 55 | 138 | 265 | 734 | 37 | 137 | 11 | 546 | 65 | 12 |
| 8 | 167 | 29 | 229 | 510 | 87 | 690 | 71 | 75 | 132 | 10 |
| 9 | 21 | 95 | 207 | 659 | 146 | 78 | 7 | 544 | 142 | 101 |

These values are obtained after running the models on entire dataset and combining them using majority voting.

# 3 Conclusion

1. All the above models that are trained support the **"No Free Lunch" theorem** which states that there is no one model that works best for every problem where assumptions of a great model for one problem may not hold for another problem which is evident from the above results.
   Hence depending on the problem, it is important to assess the trade-offs between *speed, accuracy*, and *complexity* of different models and algorithms and find a model that works best for that particular problem.
2. It is observed that the neural network algorithm gives best performance for both USPS and MNIST dataset. The confusion matrix of neural network has more values in its diagonal which means that the more of the actual and predicted values are the same when compared to those in other models.
3. By combining the results of the individual classifiers, using a classifier combination method such as majority voting, it is observed that the Neural network alone gives the best accuracy on both the datasets.