CSE 574: Introduction to Machine Learning

Amlan Gupta
#50288686
amlangup@buffalo.edu

Abstract

The project requires to us to apply four different classification methods (Logistic Regression, Multilayer Perceptron Neural Network, Random Forest and Support Vector Machine) to recognize 28×28 grayscale handwritten digit images and identify them as digits among 0, 1, 2, ... , 9. Moreover, an ensemble method needs to be implemented for the classifiers to combine and make a final decision.

1 Datasets and Data Preperation

1.1 MNIST

The MNIST database is a large database of handwritten digits that is commonly used for training various image processing systems. It was created by National Institute of Standards and Technology. The database contains 70,000 sample images.

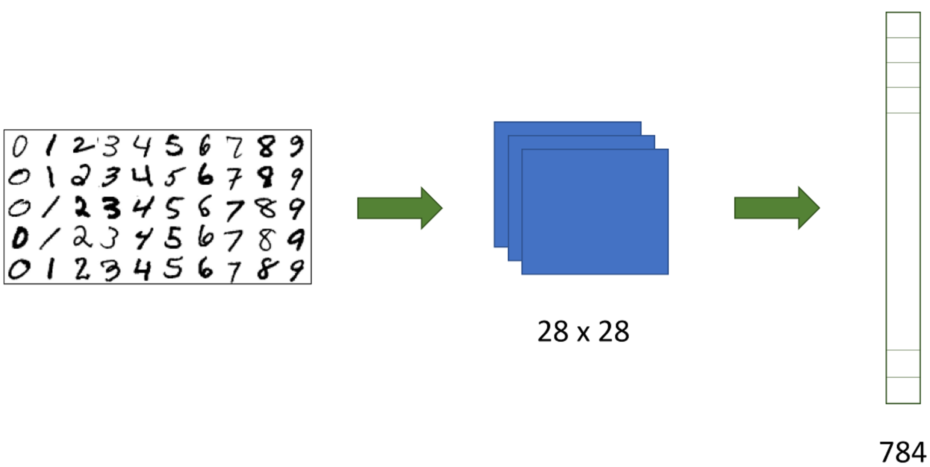


Figure 1: Creating Feature map from MNIST database

As the images were centered in a 28x28 image by computing the center of mass of the pixels, to create a feature matrix we need to flatten the 28x28 image to 784x1 feature matrix. We will be generating training set consists of 50,000 images (50000 x 784 feature matrix) and validation, testing set of 10,000 images each (10000x784 matrix).

1.2 USPS

USPS provided dataset will be used as a testing set in this project. It provides 19,999 images, where each digit type consists of 2000 images. Since we will be using this as the testing dataset, we have to use the same process used for MNIST dataset. USPS dataset needs to be resized to 28x28 images and and flatten the same way to create 784x1 feature matrix. Hence, feature matrix will be of 19999 x 784 dimensions.

2 Classification Models

2.1 Multinomial Logistic Regression using Mini-batch Stochastic Gradient Descent

2.1.1 Method of Operation

For a multi-class classification problem like handwritten digit recognition, we need to treat it as combination of multiple binary classification problem. It means we will be solving 10 different binary classification problem and merge it using softmax function to output probabilities over 10 classes.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \text{softmax}\begin{pmatrix} W_{1,1}x_1 + W_{1,2}x_2 + W_{1,3}x_3 + b_1 \\ W_{2,1}x_1 + W_{2,2}x_2 + W_{2,3}x_3 + b_2 \\ W_{3,1}x_1 + W_{3,2}x_2 + W_{3,3}x_3 + b_3 \end{pmatrix}$$

Figure 2: Yields probabilities for multiple classes using multiple sets of weights

For this problem we will be needing weights of dimension 10 x 784 dimension. In mini-batch Stochastic Gradient Descent, we select a batch of size b (b x 784 dimension matrix) from shuffled training feature matrix. After multiplying weights with the feature matrix we z which needs to passed to softmax function to normalize and product probabilities for 10 classes.

softmax(z) = $\frac{e^z}{\sum_{i=1}^{10} e_i^z}$

One hot encoding was performed on the target set of 50000x1 to trasform it to a matrix of 50000x10. The encoding transforms categorical features to a format that works better with classification and regression algorithms. We find the gradient of the error function $\nabla_{w_j} E(x)$ = $(y_j - t_j)$x

and update the weights accordingly. $\mathbf{w}^{r+1}$ = $\mathbf{w}^r$ - $\eta \sum_{i=1}^{m} \nabla_{w_j} E(z_i)$

where η is learning rate and z_i are datapoints from the mini-batch.

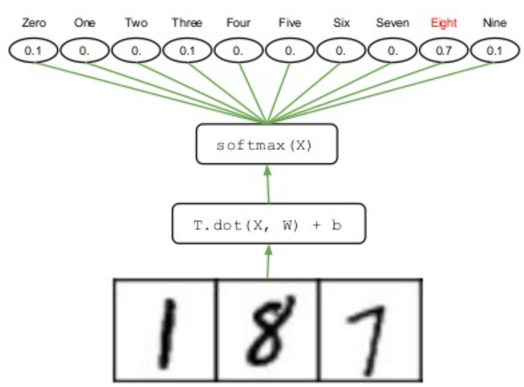


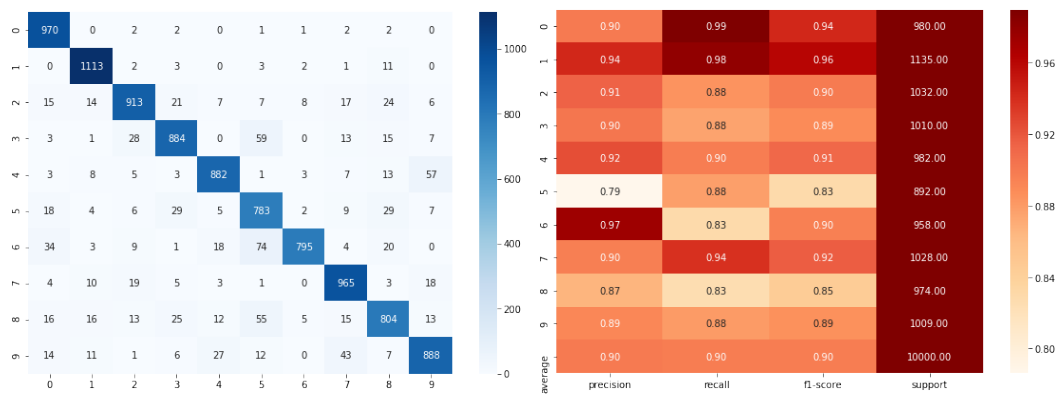
Figure 3: Generating probalities using softmax for MNIST dataset

After training the model, we can feed it n images which will generate a probability matrix of nx10 dimensions. The model decides, the image belongs to class which has the highest probability.

2.1.2 Optimal Hyper-parameters

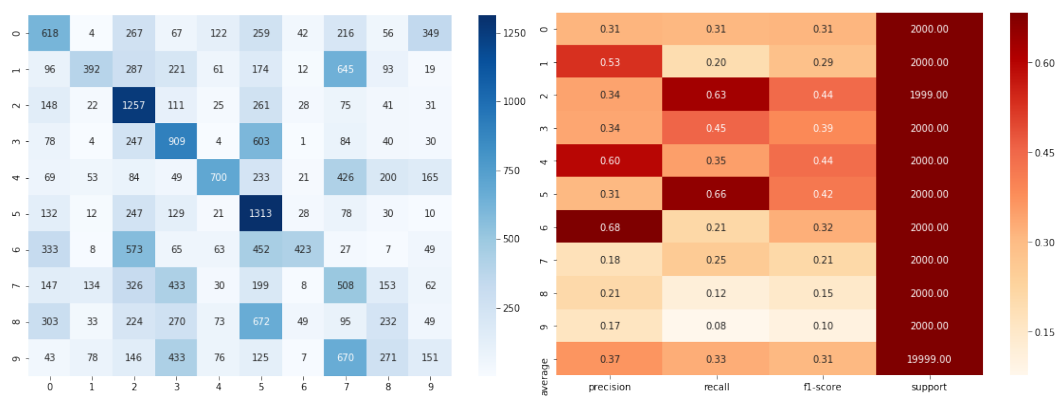
Hyper parameter	Value
Learning Rate	0.05
Mini Batch	50

2.1.3 Results



(a) Confusion Matrix (b) Classification Report

Figure 4: Report for MNIST Testing set (zoom in for clearer view)



(a) Confusion Matrix (b) Classification Report

Figure 5: Report for USPS dataset (zoom in for clearer view)

Data Set	Accuracy
MNIST Training	89.44
MNIST Validation	90.51
MNIST Testing	89.97
USPS Testing	32.51

2.1.4 Observation

The logistic regression model using mini-batch stochastic gradient descent has given a satisfactory result with accuracy of around 90%.

1. MNIST: The model performed best to recognize 1.
2. MNIST: The model could recognize most of the images for 5, however given a huge number of false positive. As a result precision decreased.
3. USPS: The number 2 and 5 was recognized the most times, however high number of false positives decreases the overall precision for this.
4. The model did not work as good with USPS dataset, Hence proved **No Free Lunch Theorem** theorem, which states that, no optimization technique (algorithm/heuristic/meta-heuristic) is the best for the generic case and all special cases (specific problems/categories). No solution therefore offers a "short cut".

2.2 Deep Neural Network

2.2.1 Method of Operation

Neural Network is loosely modeled after the neuronal structure of the mammalian cerebral cortex but on much smaller scales. A neural net consists of thousands or even millions of simple processing nodes(neurons) that are densely interconnected. Most of today's neural nets are organized into layers of nodes, and they're "feed-forward," meaning that data moves through them in only one direction. An individual node might be connected to several nodes in the layer beneath it, from which it receives data, and several nodes in the layer above it, to which it sends data.

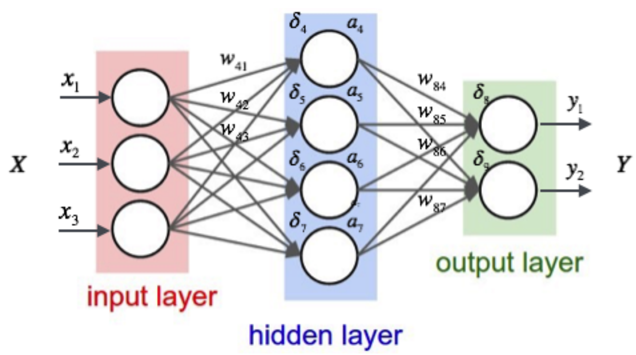


Figure 6: Neural Network (**?**)

The nodes or neurons of the input layer is passive. All they do is pass the data received from input to the next layer. Nodes in Hidden layer and output layer are active.

The values entering a hidden node are multiplied by weights, a set of predetermined numbers stored in the program. The weighted inputs are then added to produce a single number.

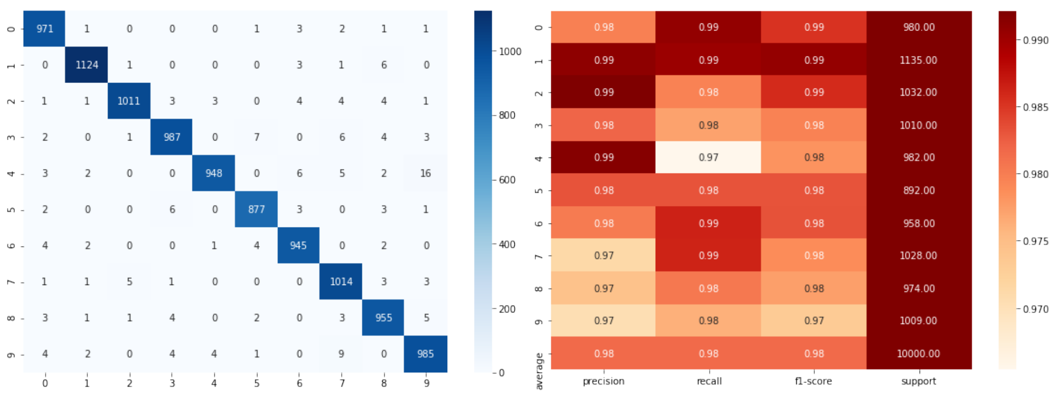
Neural networks can have any number of layers, and any number of nodes per layer. Most applications use the three-layer structure with a maximum of a few hundred input nodes.

2.2.2 Optimal Hyper-parameters

Hyper parameter	Value
Dense Layers	3
Nodes	512, 256, 10
Optimizers	adam
Loss Function	categorical_crossentropy
Epochs	10000
Model batch size	128
tb batch size	32
Early Patience	15
Activation	relu, relu, softmax

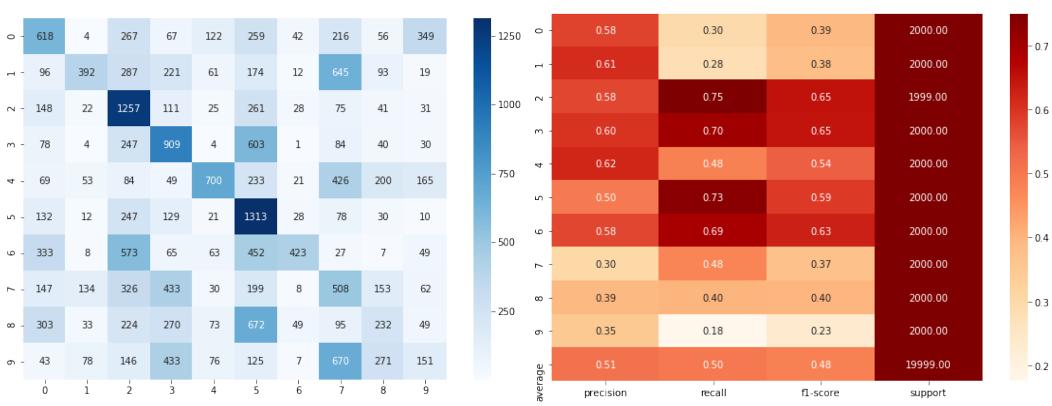
2.2.3 Results

Data Set	Accuracy
MNIST Training	99.92
MNIST Validation	98.28
MNIST Testing	98.17
USPS Testing	49.95



(a) Confusion Matrix (b) Classification Report

Figure 7: Report for MNIST Testing set (zoom in for clearer view)



(a) Confusion Matrix (b) Classification Report

Figure 8: Report for USPS dataset (zoom in for clearer view)

2.2.4 Observation

Out of all the classifiers Neural Network worked best. It was able to recognize most of the digits in MNIST dataset correctly and generated very few false positives or negatives.

1. MNIST: The model performed best to recognize 1, 2 and 4.
2. USPS: The number 2 and 3 was recognized the most times, however high number of false positives decreases the overall precision for this.
3. The model did not work as good with USPS dataset, Hence proved **No Free Lunch Theorem** theorem, which states that, no optimization technique (algorithm/heuristic/meta-heuristic) is the best for the generic case and all special cases (specific problems/categories). No solution therefore offers a "short cut".

2.3 Support Vector Machine

2.3.1 Method of Operation

A Support Vector Machine is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimentional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side.

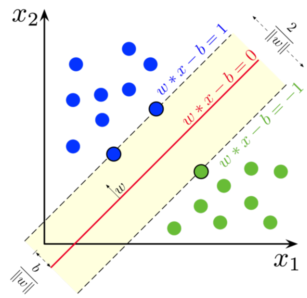


Figure 9: Support Vector Machine

In this project the support vector machine algorithm will try to find a hyperplane in an 784-dimensional space that distinctly classifies the data points.

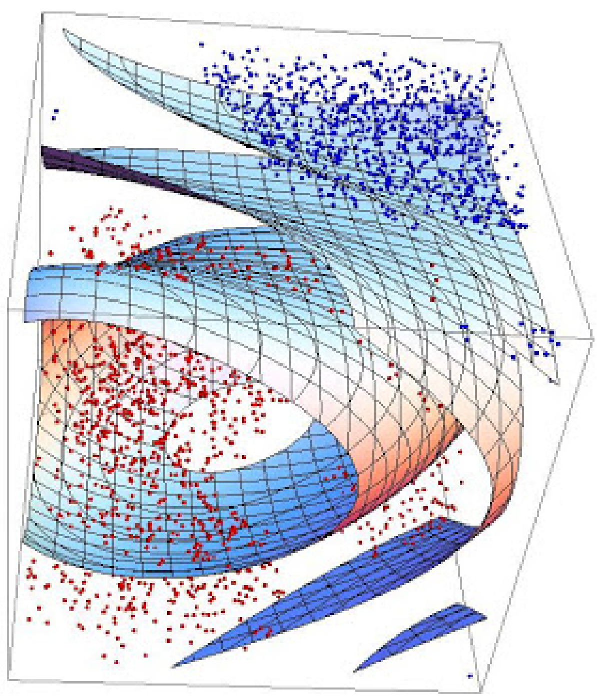


Figure 10: Support Vector Machine in Higher Dimension

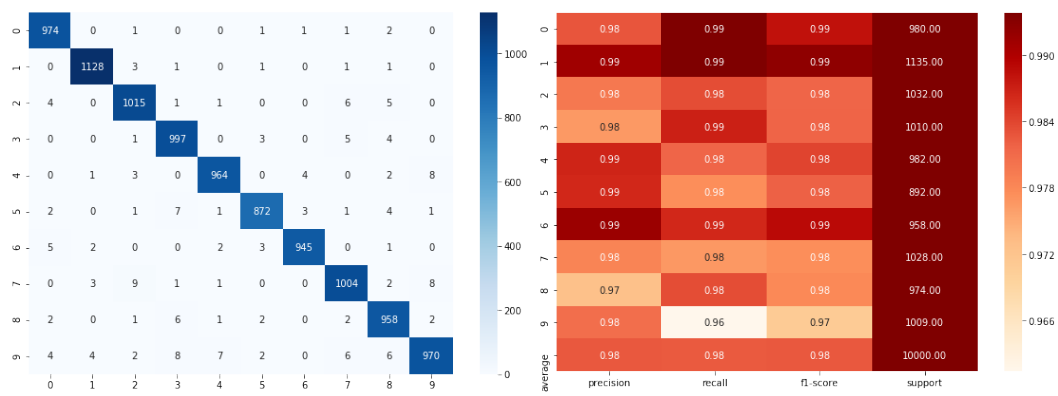
2.3.2 Optimal Hyper-parameters

Compared to the hyper-parameters suggested in the question, with following parameters the model trained faster and given better accuracy.

Hyper parameter	Value
Kernel	rbf
Gamma	0.05
C	2.0

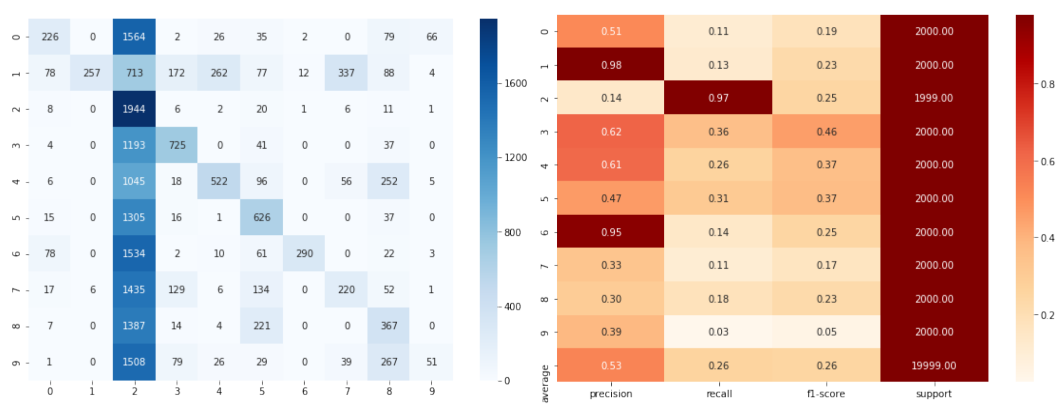
2.3.3 Results

Data Set	Accuracy
MNIST Training	99.99
MNIST Validation	98.35
MNIST Testing	98.27
USPS Testing	26.14



(a) Confusion Matrix (b) Classification Report

Figure 11: Report for MNIST Testing set (zoom in for clearer view)



(a) Confusion Matrix (b) Classification Report

Figure 12: Report for USPS dataset (zoom in for clearer view)

2.3.4 Observation

Though slower than other classifiers, support vector machine worked incredibly well. Out of all the classifiers svm's precision is highest.

1. MNIST: The model performed best to recognize 1, 4, 5 and 6.

2. MNIST: The model could recognize most of the images for 5, however given a huge number of false positive. As a result precision decreased.

3. USPS: The number 2 was recognized the most times, however a huge number of false positives decreases the overall precision for this significantly.

4. USPS: model predicted a huge number of false-negatives for number 1.

5. The model made significantly bad prediction with USPS dataset with accuracy of only 26%, Hence proved **No Free Lunch Theorem** theorem, which states that, no optimization technique (algorithm/heuristic/meta-heuristic) is the best for the generic case and all special cases (specific problems/categories). No solution therefore offers a "short cut".

2.4 Random Forest

2.4.1 Method of Operation

Random Forest is a supervised learning algorithm. Just like a forest that consists of trees, random forest is the combination of decision trees. This model ensembles decisions given by individual trees using methods like bagging, majority voting etc.

Random Forest adds additional randomness to the model, while growing the trees. Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features. This results in a wide diversity that generally results in a better model.

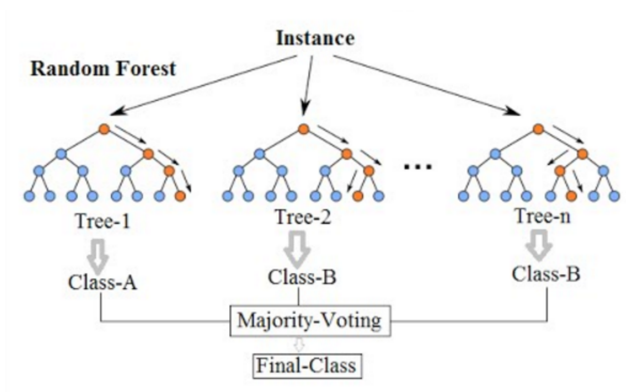


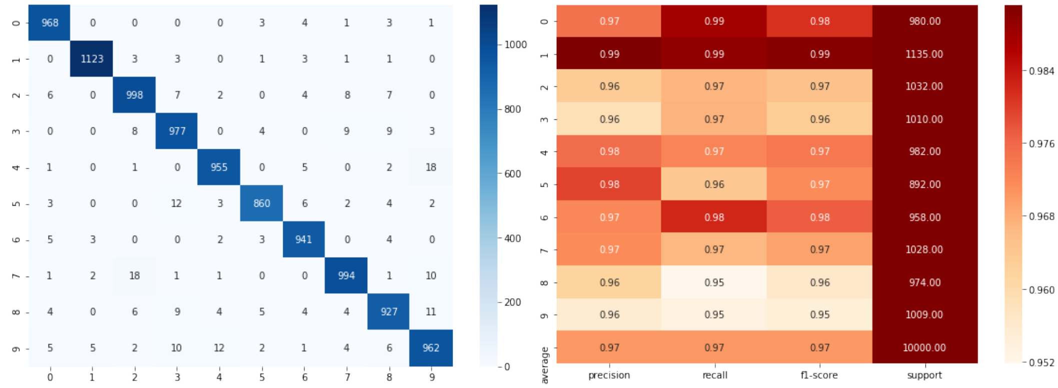
Figure 13: Random Forest Algorithm

2.4.2 Optimal Hyper-parameters

Hyper parameter	Value
Estimators	1000

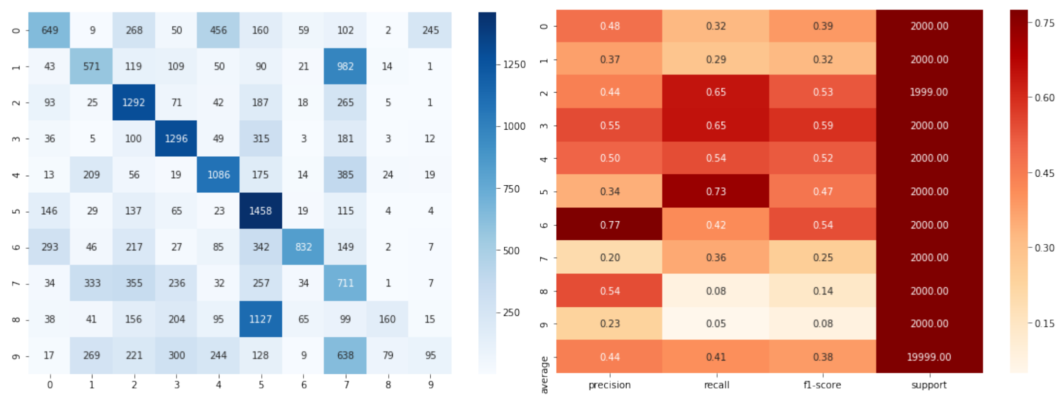
2.4.3 Results

Data Set	Accuracy
MNIST Training	100
MNIST Validation	97.34
MNIST Testing	97.05
USPS Testing	40.75



(a) Confusion Matrix (b) Classification Report

Figure 14: Report for MNIST Testing set (zoom in for clearer view)



(a) Confusion Matrix (b) Classification Report

Figure 15: Report for USPS dataset (zoom in for clearer view)

2.4.4 Observation

1. Random forest was able to fully recognized the training set, so this mode with this many estimator may made it overfit.
2. MNIST: The model performed best to recognize 2.
3. USPS: The model could recognize most of the images for 5, however given a huge number of false positive. As a result precision decreased.
4. The model made bad prediction with USPS dataset, Hence proved **No Free Lunch Theorem** theorem, which states that, no optimization technique (algorithm/heuristic/meta-heuristic) is the best for the generic case and all special cases (specific problems/categories). No solution therefore offers a "short cut".

2.5 Ensemble using Soft Voting

2.5.1 Method of Operation

Using 4 different classifiers we have predicted 4 different results for a single data point. Now using ensemble method we would like to combine the predication and come up with a model with better accuracy.

In soft voting, we predict the class labels by averaging the class-probabilities found from different classifiers. This way, if a classifier has more conviction on a desicion, it's given weight in the ensemble operation instead of voting out by other 'less-sure-decisions'

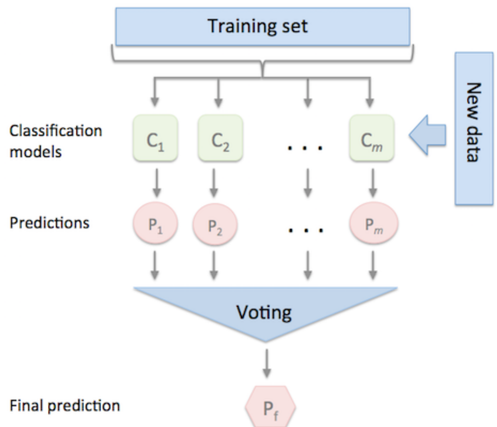
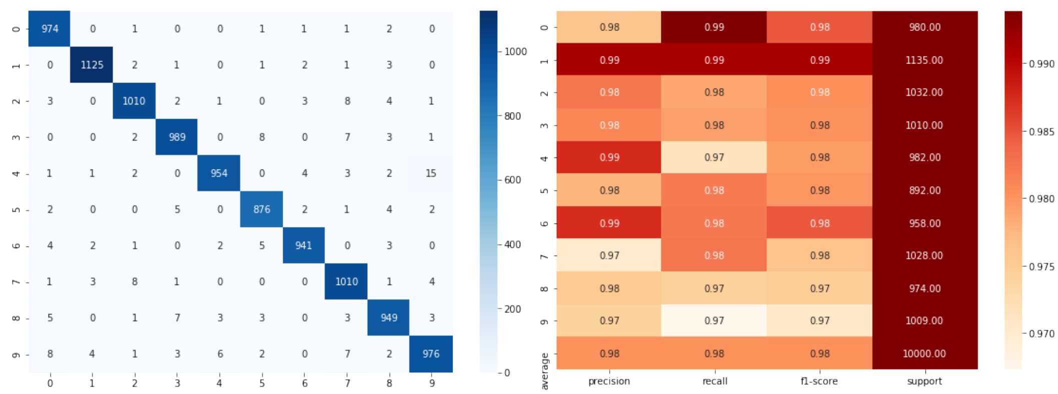


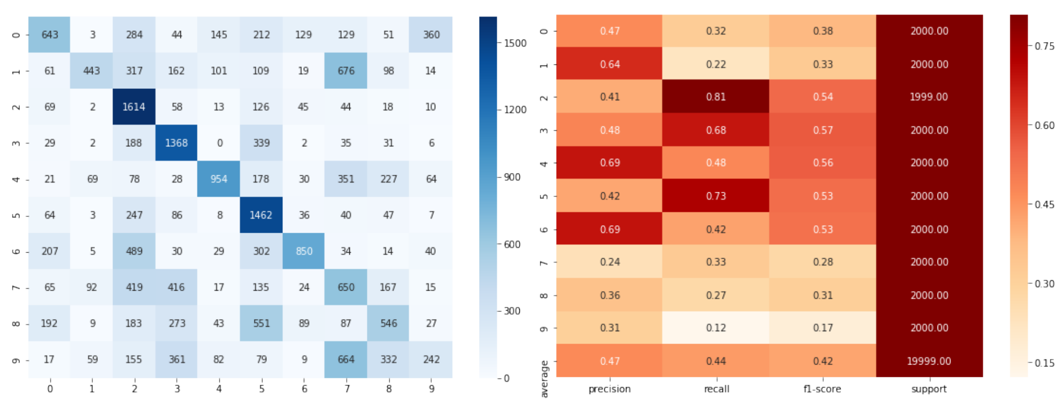
Figure 16: Ensemble using soft voting

2.5.2 Results



(a) Confusion Matrix (b) Classification Report

Figure 17: Report for MNIST Testing set (zoom in for clearer view)



(a) Confusion Matrix (b) Classification Report

Figure 18: Report for USPS dataset (zoom in for clearer view)

Data Set	Accuracy
MNIST Testing	98.04
USPS Dataset	43.86

2.5.3 Observation

1. MNIST: The model performed best to recognize 1, 4, 6.
2. USPS: The model could recognize most of the images for 2, however given a huge number of false positive. As a result precision decreased.
3. The model made bad prediction with USPS dataset, Hence proved **No Free Lunch Theorem** theorem, which states that, no optimization technique (algorithm/heuristic/meta-heuristic) is the best for the generic case and all special cases (specific problems/categories). No solution therefore offers a "short cut".

3 Conclusion

The following graph represents the accuracy of the classifiers:

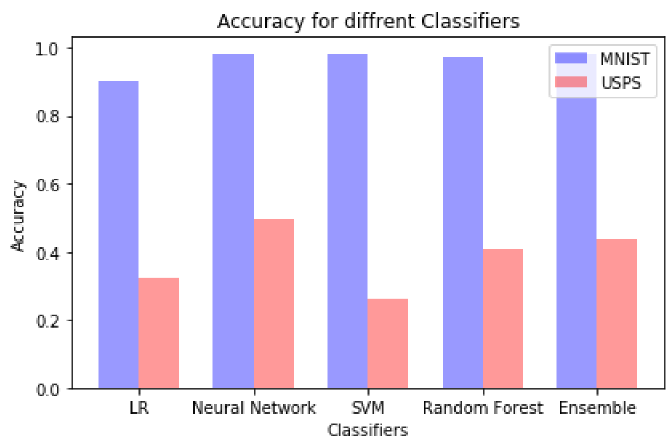


Figure 19: Accuracy for different classifiers

1. Considering both dataset the ensemble method could not outperform Neural Network, which consistently gave better result for MNIST and USPS dataset.
2. Including logistic regression in the ensemble method decreased the overall accuracy of it. We can conclude that, ensemling should always be performed on classifiers that are performing equally good.
3. Using soft voting instead of hard voting worked better in this case, as neural network was able to perform recognition with more conviction, as a result decreasing the effect of logistic regression.
4. All the classifiers consistanty proved No Free Lunch theorem.