CSE574 Introduction to Machine Learning

Project 3

Classification of Grayscale Handwritten Digit Image Using Logistic Regression, Neural Network, Random Forest and SVM Classifiers

Project Report

Submitted By

Karan Manchandia

Abstract

This project aims to apply the concept of machine learning in a classification task. The task would be to classify 28*28 grayscale handwritten digit image and identify them as digit among 0,1,2….9. There are two given datasets, one is MNIST data set and another is USPS data set. We will train, validate and test our model of the four classifiers namely Logistic Regression, Neural Network, Random Forest and SVM (Support Vector Machine) using the MNIST data set. We will be only using USPS data set to test our model. The MNIST data set contains 60,000 training images and 10,000 testing images. We use USPS handwritten digit images as the testing data to test the generalization power of our model. In this project we will also implement the ensemble of the four classifiers. Finally, the result of all the classifiers will be combined to get the final result.

Introduction

There are four classifiers using which we will solve this problem:

1. By using Logistic Regression Solution
2. By using Neural Network Solution
3. By using Random Forest Solution
4. By using SVM (Support Vector Machine) Solution

   In the end we will do ensemble of these four classifiers using Majority Voting.

Steps Involved in Project Implementation:

1. **Extract feature values and labels from the data**: Process the MNIST data into a Numpy array that contains the feature vectors and a Numpy array that contains the labels.

2. **Data Partition**: The given MNIST data set is already partitioned into training data set and testing data set. We will be using the training data set to train our model.

3. **Train model parameter**: We will be training each of our models for different hyperparamaters such as the number of iterations, learning rate, number of hidden layers etc.

4. **Tune hyper-parameters**: In this step, we will be tuning our hyperparameters and will try to find out the hyperparameters for each model where we get the best training auracy.

5. **Evaluate testing sets**: Test the trained models on both MNIST test set and USPS data. Discuss your findings.

Logistic Regression Solution:

Logistic Regression is defined as a classification algorithm that can be used to map each observation to a discrete set of classes. Logistic Regression uses a logistic sigmoid function to return a value which can be mapped to various classes.

Why we use Sigmoid Function in logistic regression?

The sigmoid function is used to map a particular real valued output to a output between zero and one i.e. in order to map a predicted value to probability we use the sigmoid function. We use a sigmoid function to map predictions to probabilities in machine learning.

The formula for sigmoid function is given by:

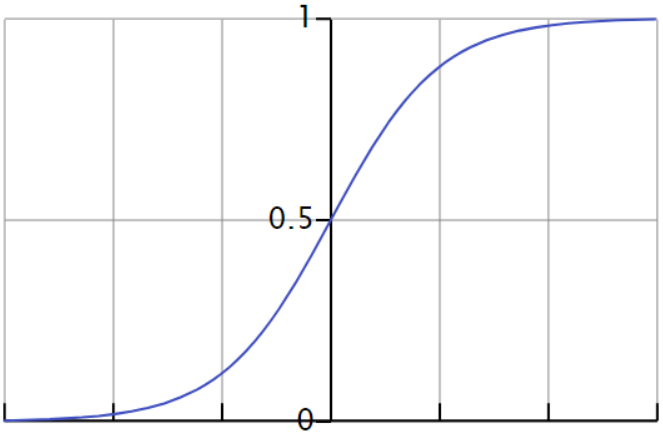
$$f(t) = \frac{1}{1+e^{-t}}$$

Here, f(t) = output between 0 and 1.

T = input the the function

e = base of the natural log

A sigmoid function is shown below:



In logistic regression, we can not use the same cost function as it was used in case of linear regression. So, we use logarithmic cost function as shown below:

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

The above equation can be rewritten as:

$$-\frac{1}{m}[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)}))]$$

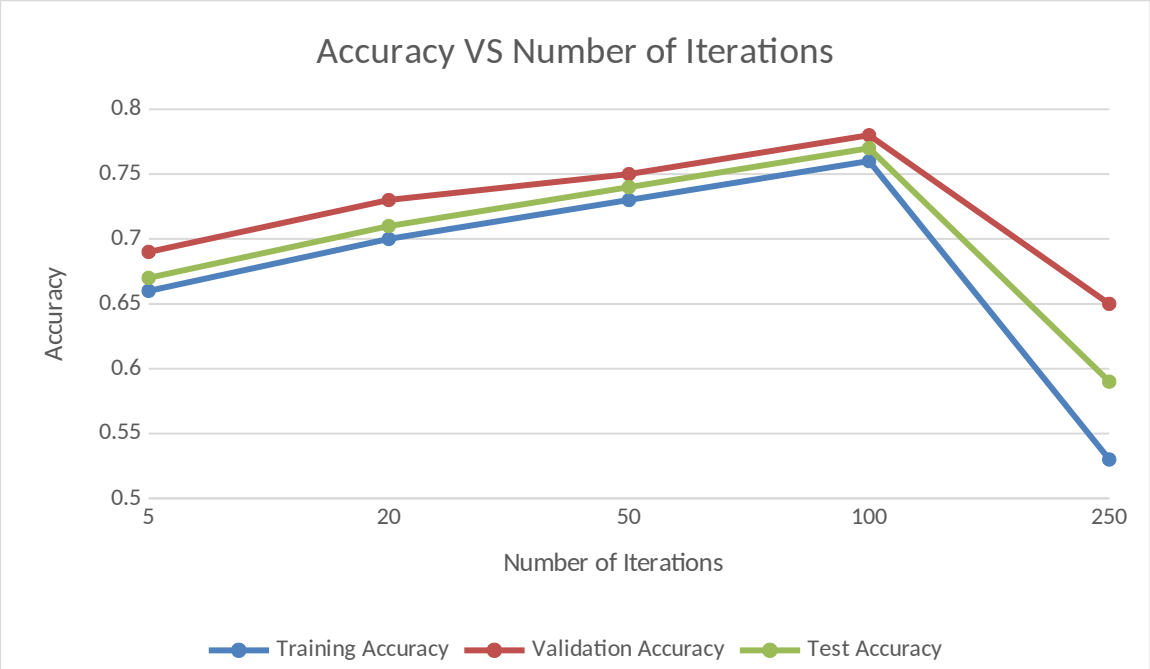
Finally, we use the equation of the type as shown below for calculating gradients:

$$\frac{\partial J}{\partial \theta_n} = \frac{1}{m} \cdot x_i \cdot [\sum_{i=1}^{m} h_i - y_i]$$

For learning rate = 0.01, varying no. of iterations and checking Training, Testing and Validation accuracy for MNIST data set

[Note: The accuracies written in all the tables below are not in percentage, multiply these values by 100 to get accuracy in percentage.]

Learning Rate	No. of Iterations	Training Accuracy	Validation Accuracy	Test Accuracy
0.01	5	0.66	0.69	0.67
0.01	20	0.70	0.73	0.71
0.01	50	0.73	0.75	0.74
0.01	100	0.76	0.78	0.77
0.01	250	0.53	0.65	0.59



It can be clearly seen in the graph above that as we increase the number of iterations accuracy in all three cases increases. But after number of iterations crosses 100, accuracy decreases. This is due to overfitting.

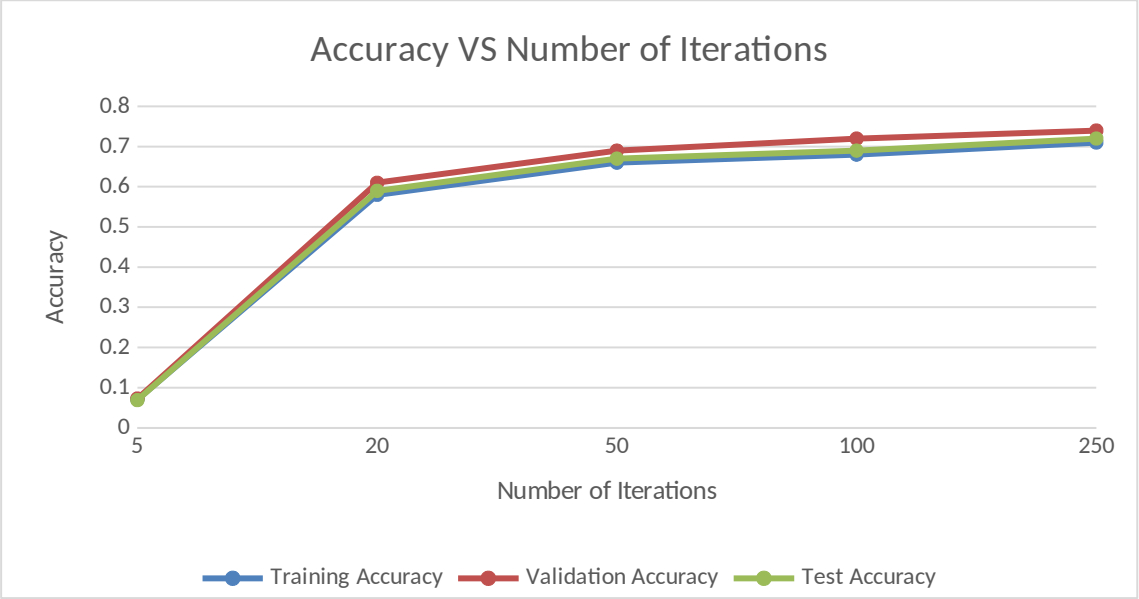
Why Overfitting? Why overfitting is bad in Machine Learning?

When we set epochs (number of iterations) to larger number without early stopping, it will cause the problem of overfitting. This will cause further classifications to be error-some. If you train your model to fit each and every data, the model would not have learnt, how to deal with data that it is not trained on.

Overfitting is bad since this does not generalize the training model and accuracy on new test data may not be very great.

For learning rate = 0.001, varying no. of iterations and checking Training, Testing and Validation accuracy for MNIST data set

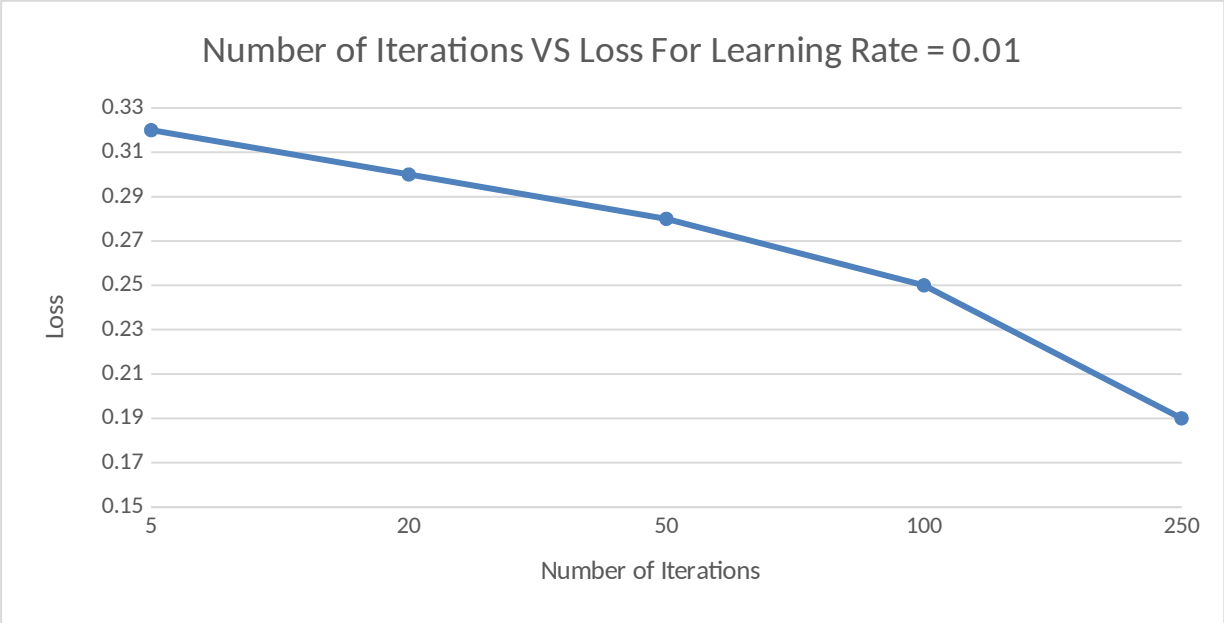
Learning Rate	No. of Iterations	Training Accuracy	Validation Accuracy	Test Accuracy
0.001	5	0.071	0.073	0.069
0.001	20	0.58	0.61	0.59
0.001	50	0.66	0.69	0.67
0.001	100	0.68	0.72	0.69
0.001	250	0.71	0.74	0.72



It can be clearly seen from the graph that accuracy increases as number of iterations increases.

For learning rate = 0.01, varying no. of iterations and checking Loss for MNIST data set

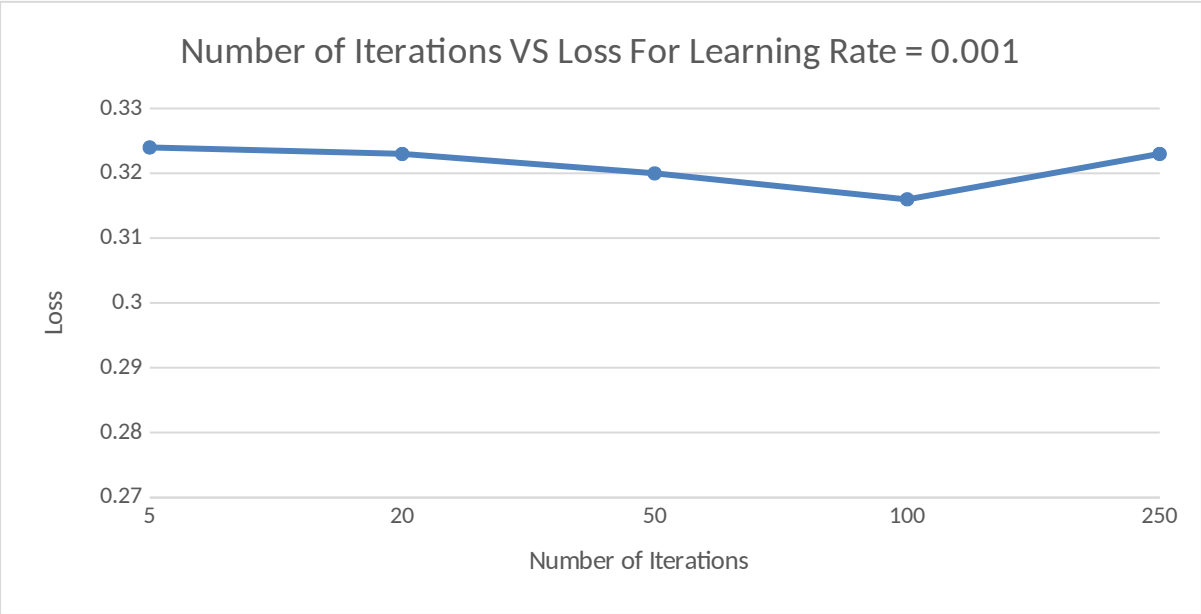
Learning Rate	No. of Iterations	Loss
0.01	5	0.32
0.01	20	0.30
0.01	50	0.28
0.01	100	0.25
0.01	250	0.19



It can be clearly seen from the graph above that loss decreases as number of iterations increases.

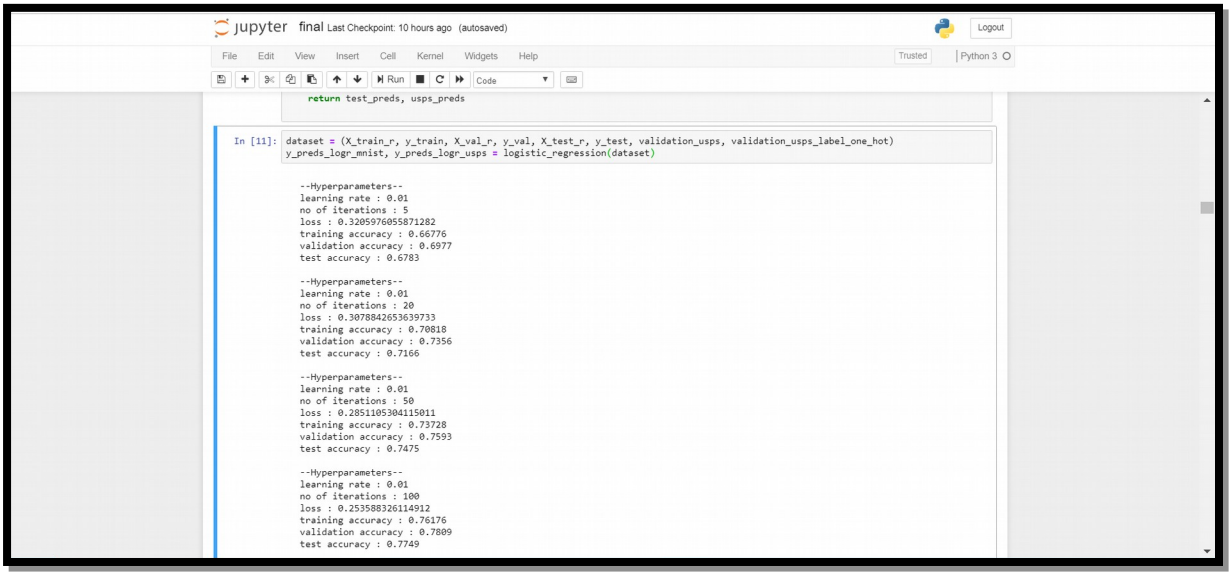
For learning rate = 0.001, varying no. of iterations and checking Loss for MNIST data set

Learning Rate	No. of Iterations	Loss
0.001	5	0.324
0.001	20	0.323
0.001	50	0.320
0.001	100	0.316
0.001	250	0.323



It can be clearly seen from the graph above that loss does not change much with change in number of iterations for learning rate = 0.001.

Screenshots for Logistic Regression Results:

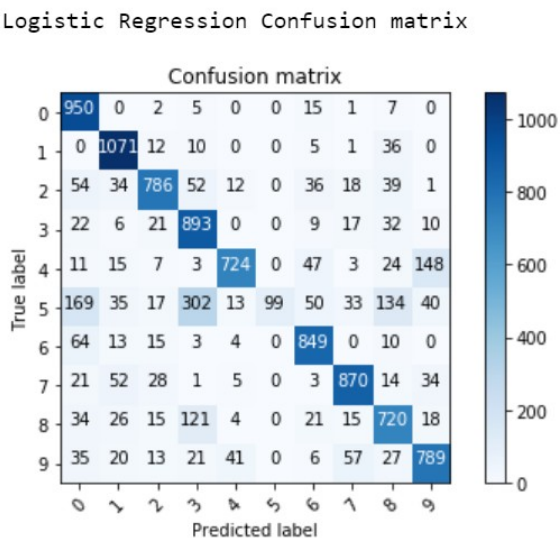


Dimentionality Table:

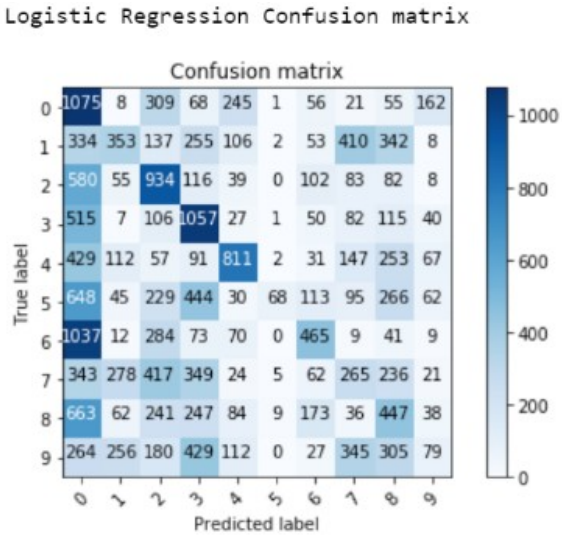
Dataset	Dimentions
MNIST Training Data Set	(50,000, 784)
MNIST Validation Data Set	(10,000, 784)
MNIST Test Data Set	(10,000, 784)

Confusion Matrices:

Logistic Regression Confusion Matrix for MNIST data set



Logistic Regression Confusion Matrix for USPS data set

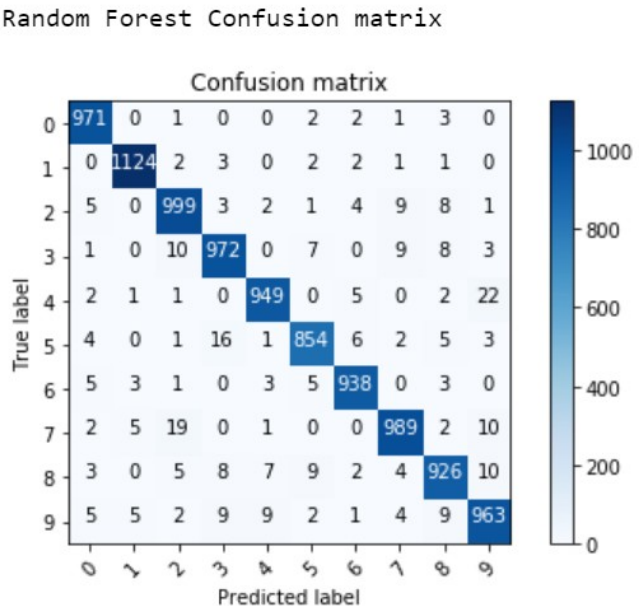


Randon Forest:

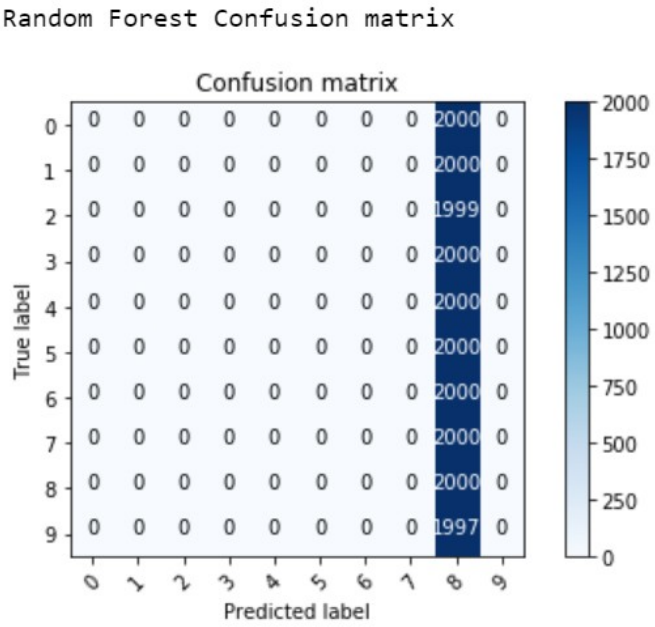
Random Forest is also called Random Decision Forest. Random Forest constructs a Random Decision Tree while we train the model. At the time of testing, it classifies the input based on multiple decision trees it constructed at Training Time. In each decision tree, a decision is made at each node in order ot reach the final classification. Random Forest can also be used in regressin problems and other tasks. We used Random Forest Grid Search for tuning Hyperparameters in this model.

Confusion Matrices:

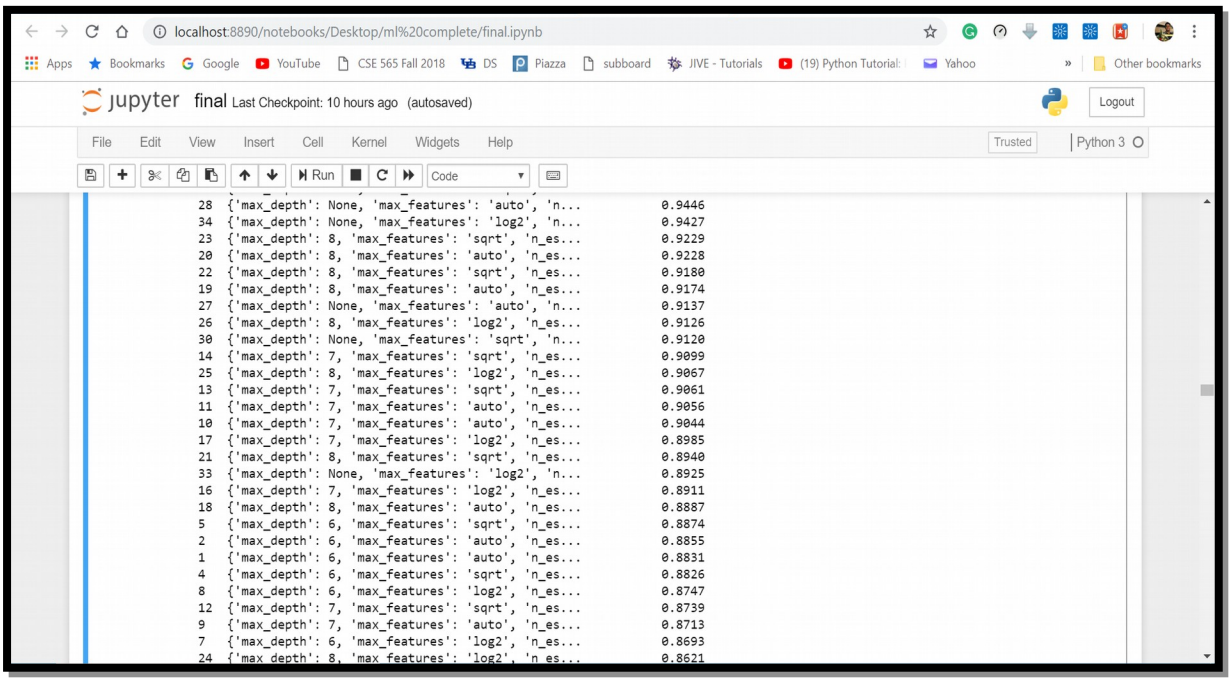
Random Forest Confusion Matrix for MNIST data set:

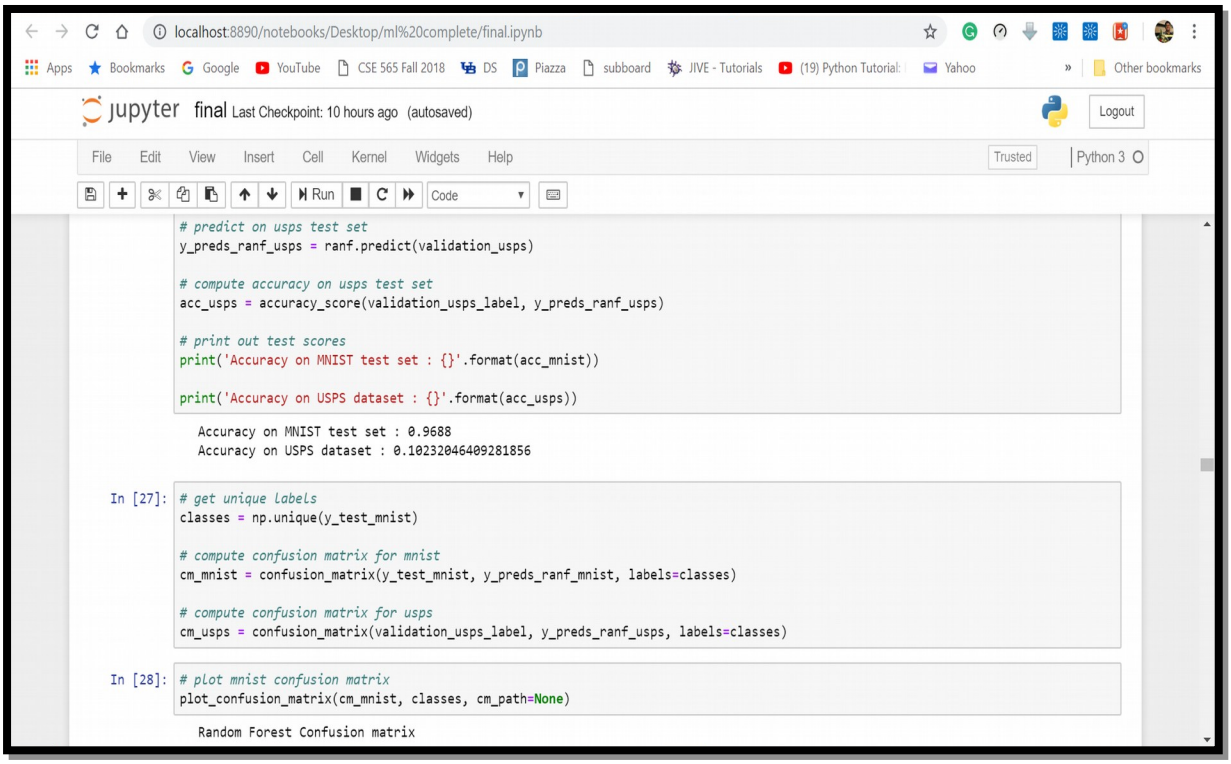


Random Forest Confusion Matrix for USPS data set:



Screenshots of Random Forest Results:





The screenshot above shows that:

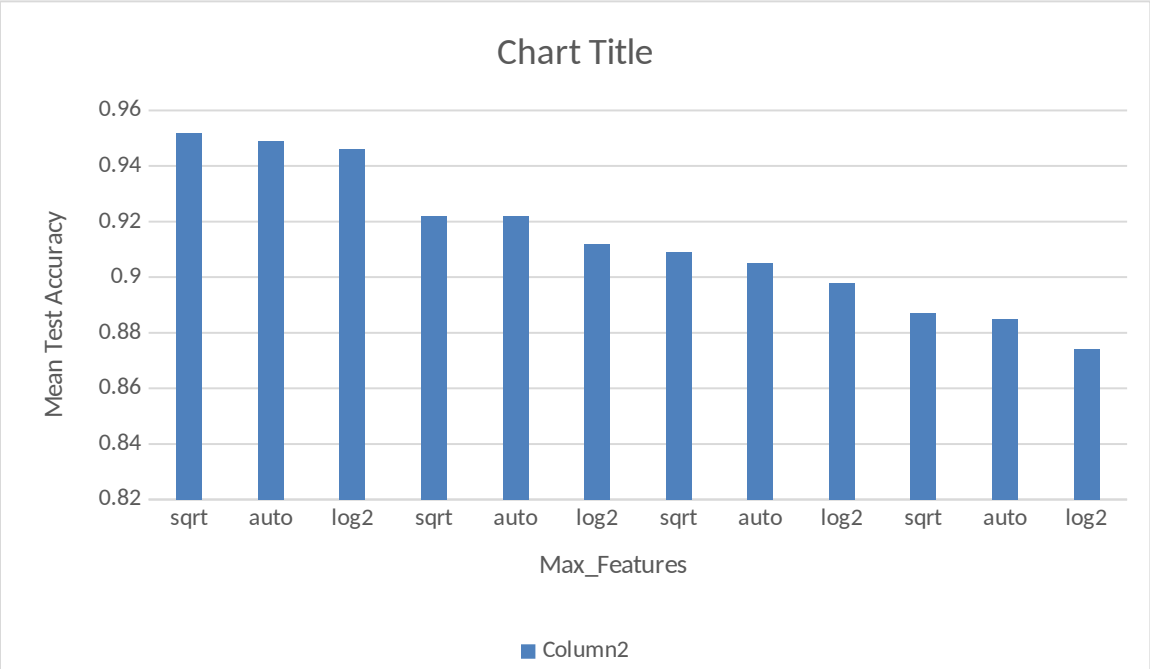
Accuracy on MNIST test set = 0.9688

Accuracy on USPS data set = 0.1023

Hyperparameters Tuning for Random Forest:

[Note: There are many hyperparameters tuned as shown in the screenshots above. Only a few results are listed in this table.]

Max_Depth	Max_features	Mean Test Accuracy
none	sqrt	0.952
none	auto	0.949
none	log2	0.946
8	sqrt	0.922
8	auto	0.922
8	log2	0.912
7	sqrt	0.909
7	auto	0.905
7	log2	0.898
6	sqrt	0.887
6	auto	0.885
6	log2	0.874

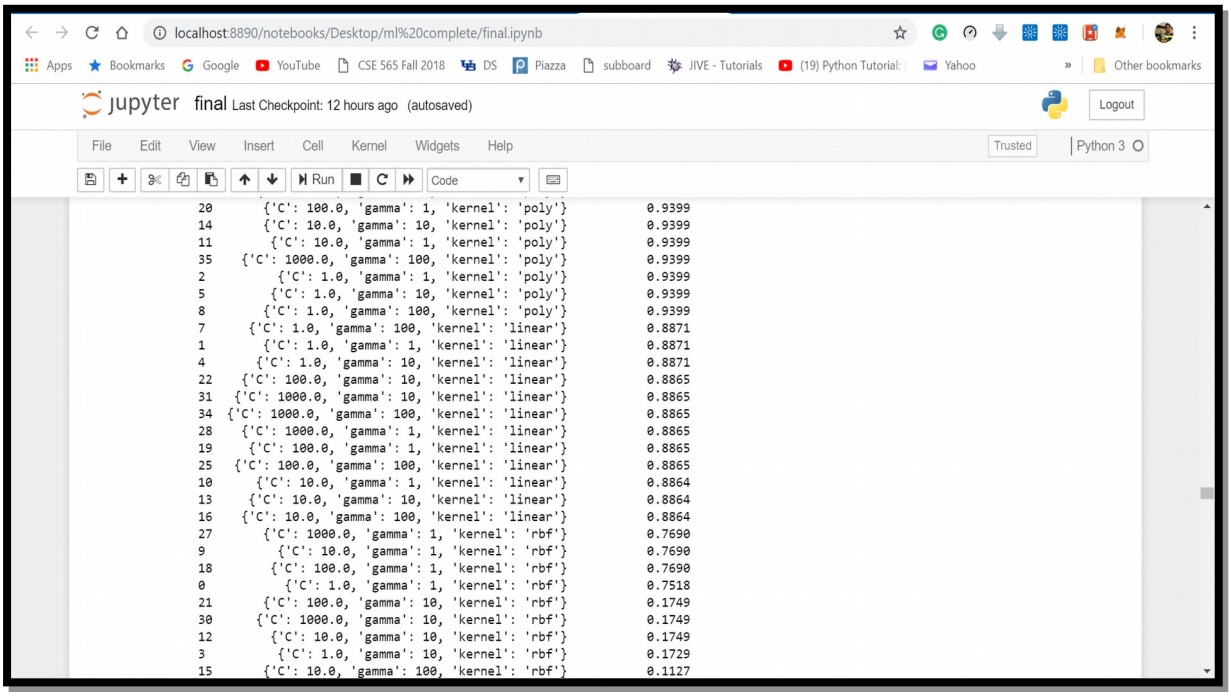


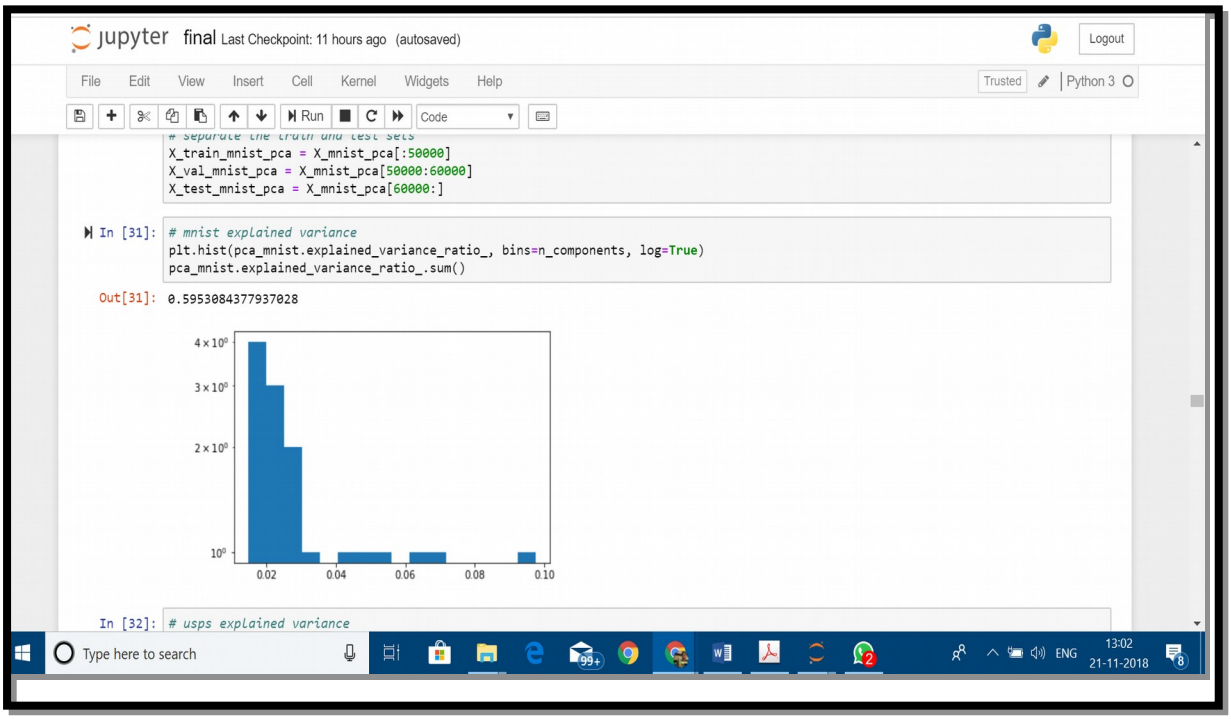
It can be clearly seen from graph and table above that accuracy decreases with decreasing depth.

SVC (Support Vector Machine):

Support Vector Machine is a supervised learning model and is used for classification and Regression analysis. SVM divides examples of separate categories by a clear gap that is as wide as possible. SVM model assigns new examples to one category orthe other. Hence SVM is a non probabilistic binery linear classifier.

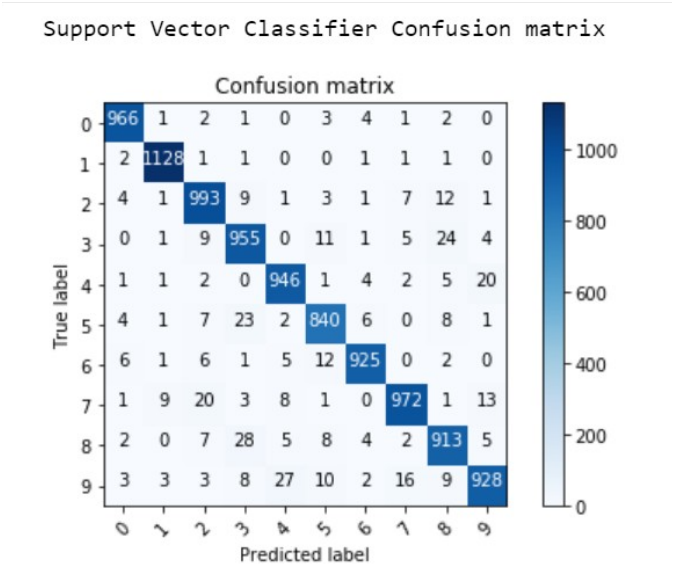
Screenshots of SVC Results:



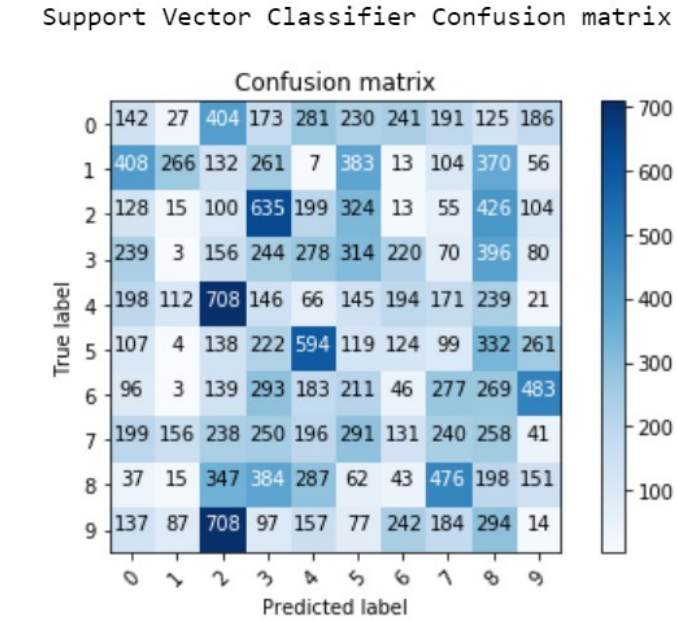


Confusion Matrices for SVC:

MNIST Confusion Matrix for Support Vector Classifier:



USPS Confusion Matrix for Support Vector Classifier:



Tuning various Hyperparameters and Getting Mean Test Accuracy:

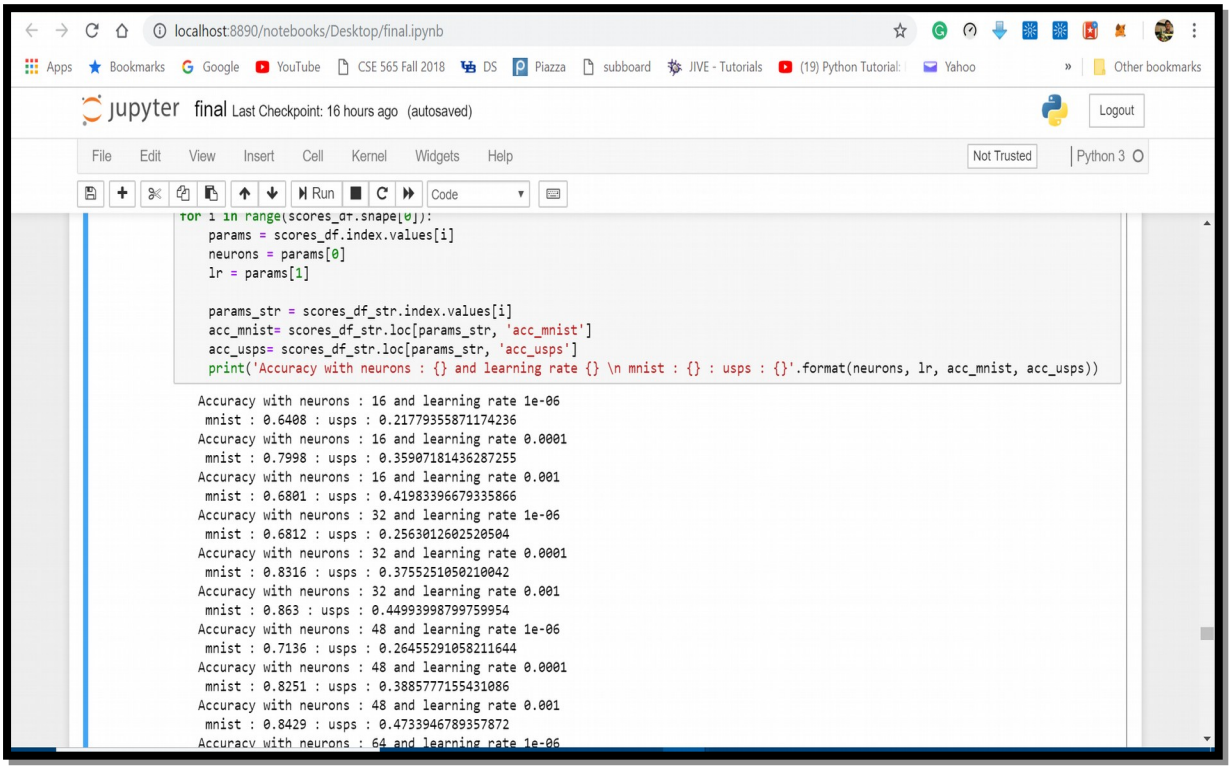
C (Penalty parameter C of the error term)	Gamma	Kerenel	Mean Test Accuracy
10.0	100	poly	0.9399
1000.0	10	poly	0.9399
100.0	100	poly	0.9399
1.0	100	linear	0.8871
100	10	linear	0.8865

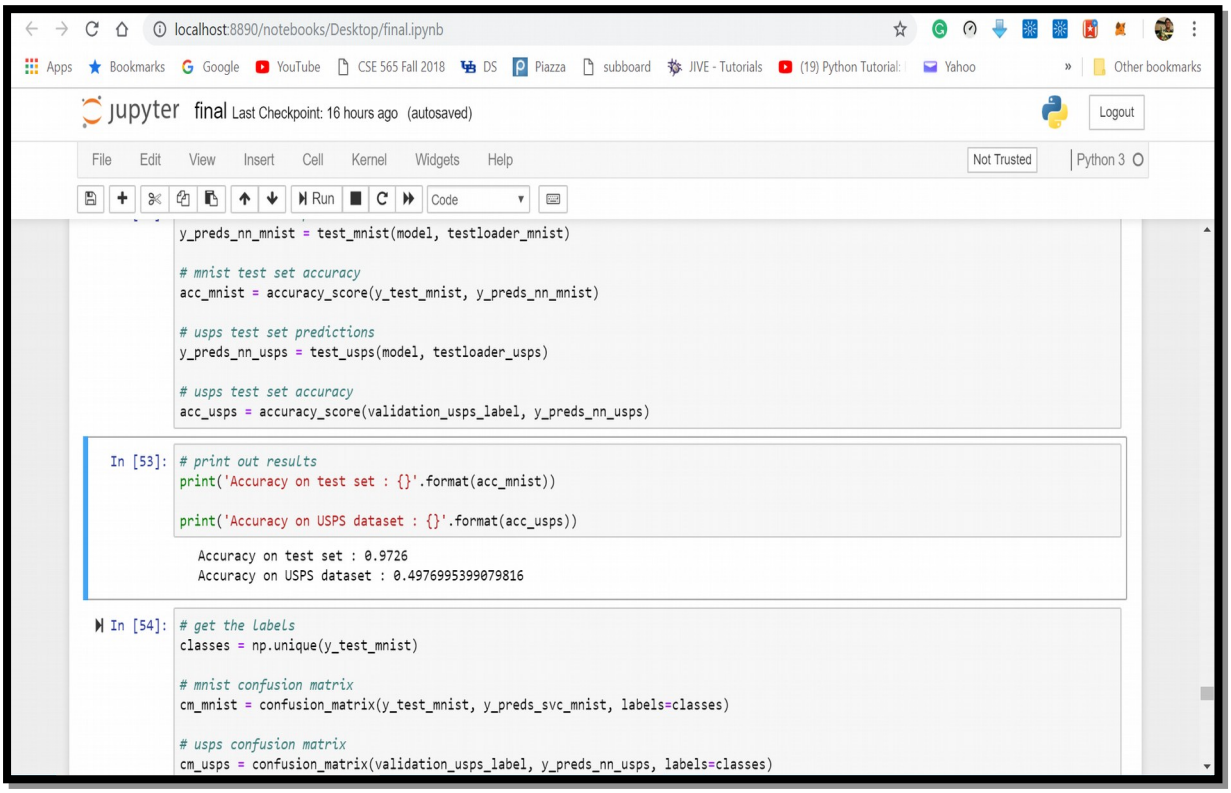
1000	10	linear	0.8865
100	1	rbf	0.7690
10	1	rbf	0.7690
1.0	1	rbf	0.7518
100	10	rbf	0.1749
10	100	rbf	0.1127

Neural Network Solution:

Neural Network is defined as a specific set of algorithm that is inspired by biological neural networks. The neural network is general function algorithm and hence can be applied to any problem that can be solved using machine learning. The figure below shows the block diagram of a two layered Neural Network Model.

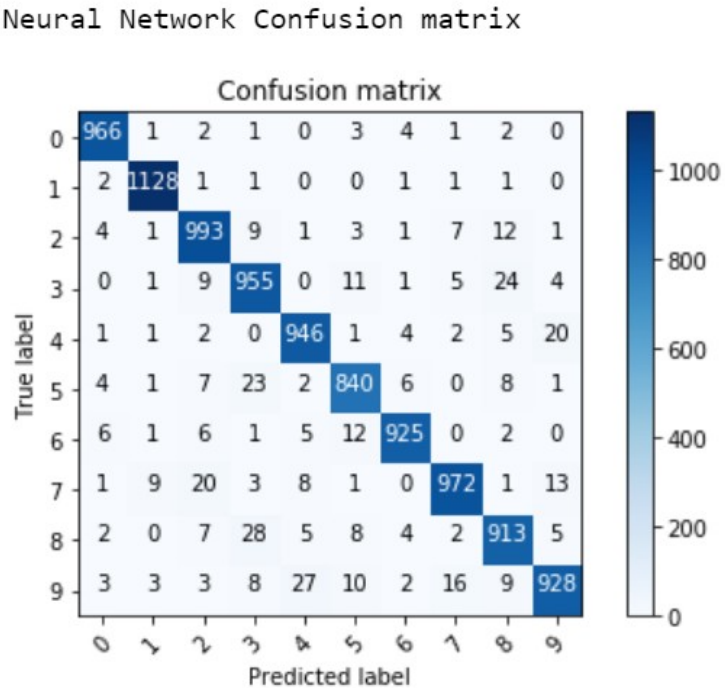
Screenshots:



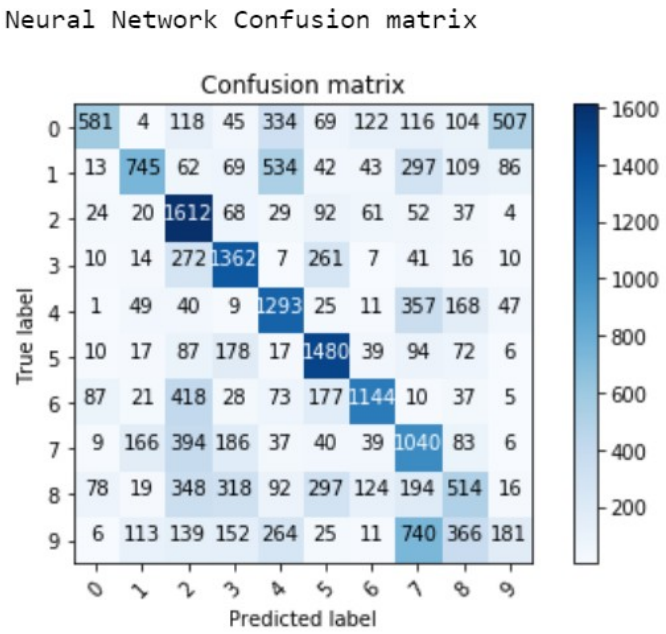


Confusion Matrices:

MNIST Confusion Matrix for Neural Network:



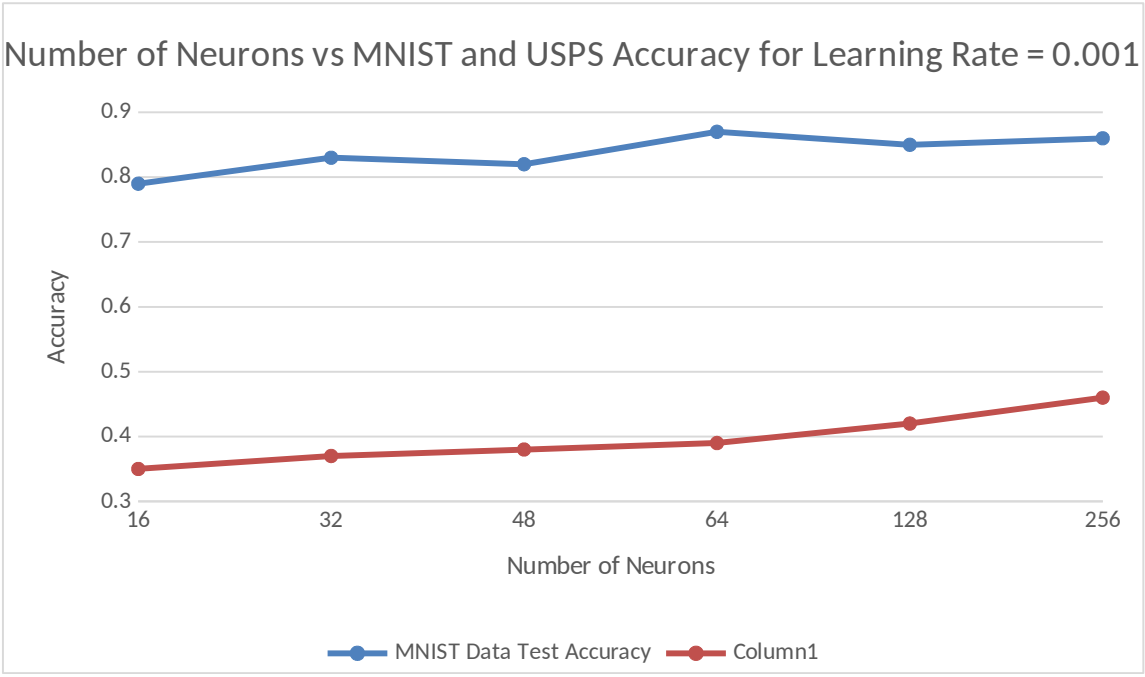
USPS Confusion Matrix for Neural Network:

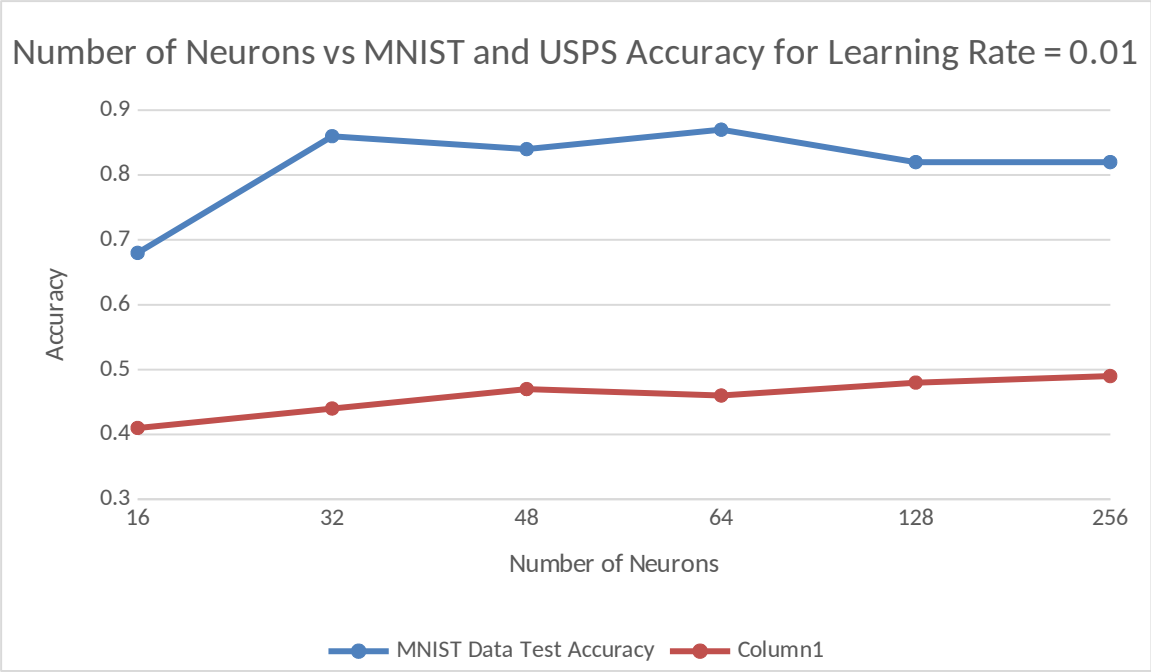


Tuning number of neurons and learning rate and plotting MNIST test accuracy and USPS test accuracy:

No. of Neurons	Learning Rate	MNIST Data Test Accuracy	USPS Data Test Accuracy
16	0.001	0.79	0.35
32	0.001	0.83	0.37
48	0.001	0.82	0.38
64	0.001	0.87	0.39
128	0.001	0.85	0.42
256	0.001	0.86	0.46
16	0.01	0.68	0.41
32	0.01	0.86	0.44
48	0.01	0.84	0.47
64	0.01	0.87	0.46
128	0.01	0.82	0.48
256	0.01	0.82	0.49

Plotting Number of Neurons vs MNIST and USPS data Accuracy:

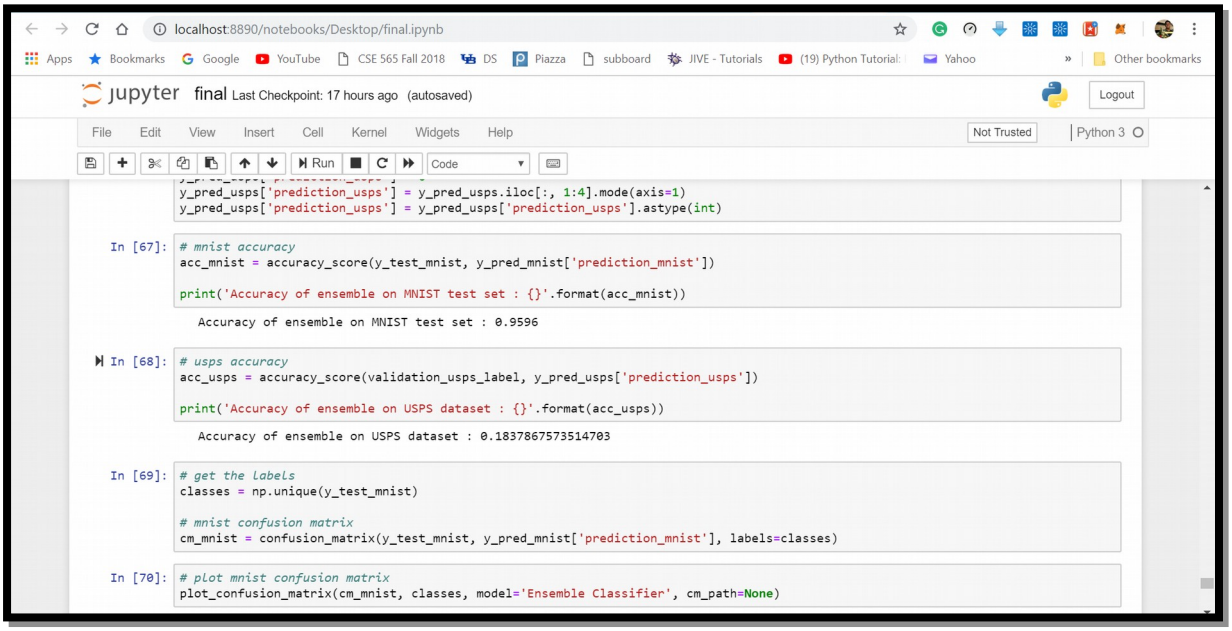




Ensemble of Classifiers (Majority Voting):

An Ensemble of classifiers can be defined as a set of classifier in which results of each Classifier is combined in some way to classify new examples. In majority voting, each training data subset is used to train a different classifier of the same type. Each Classifier is then combined by taking a majority vote of their decision. The class chosen by most of the classifiers is the ensemble decision.

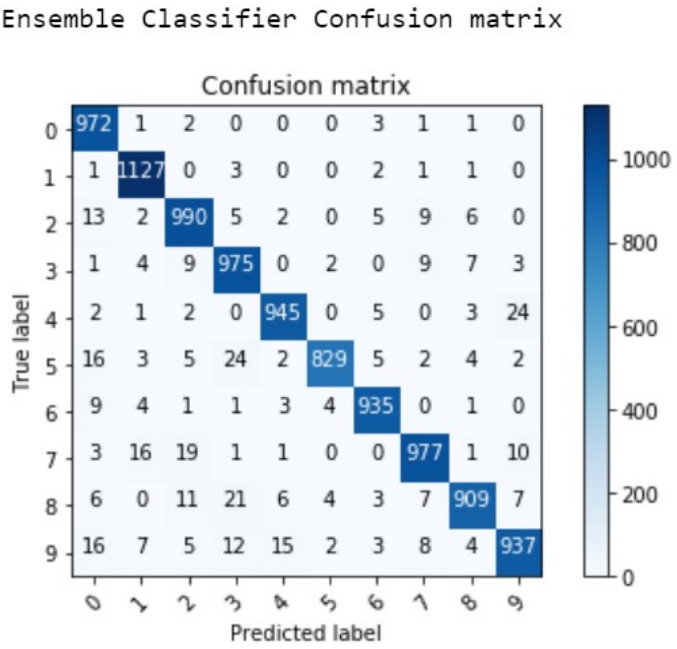
Screenshots:



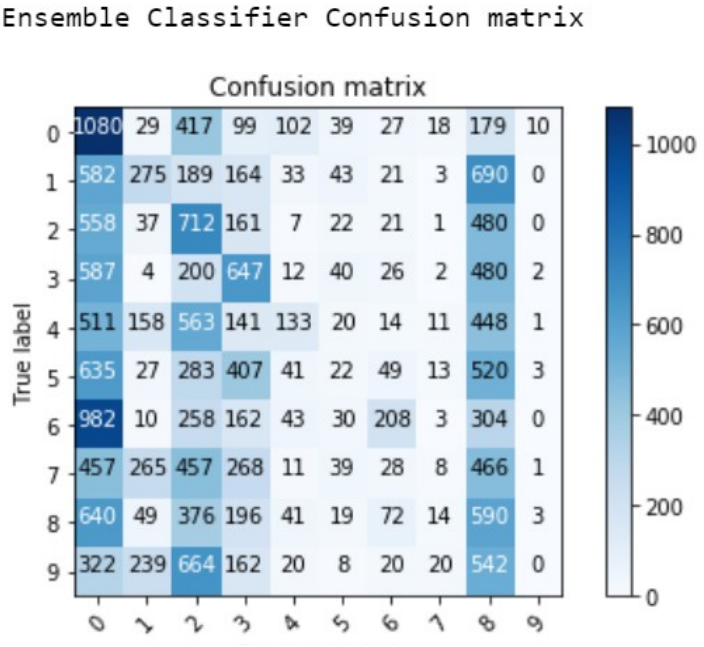
It can be clearly seen in the screenshot above that accuracy of ensemble of MNIST test set is 0.9596 and the accuracy of ensemble of USPS data set is 0.1837.

Confusion Matrices:

MNIST Confusion Matrices for Ensemble Classifier:



USPS Confusion Matrix for Ensemble Classifier:



Conclusion:

- Selection of the most optimal Hyperparameters for Logistic Regression:

learning rate : 0.01

no of iterations : 100

loss : 0.253588326114912

training accuracy : 0.76176

validation accuracy : 0.7809

usps accuracy : 0.27770554110822165

- Results for Random Forest:

Accuracy on MNIST test set = 0.9688

Accuracy on USPS data set = 0.1023

- Results for Neural Network:

Accuracy on MNIST test set : 0.9726

Accuracy on USPS dataset : 0.4976995399079816

- Maximum test accuracy for SVM is 0.9399

Questions to be Answered:

- **Do our results support "No Free Lunch" Theorem?**

The "No Free Lunch" theorem suggests that there is no one model that works best for every

problem. The assumptions of a model for one data set may not work best for another data set. In our problem, the model performed poorly on USPS data set, thus confirming the no free lunch theorem. Hence, our results support "No Free Lunch" theorem.

- **Observe the confusion matrix of each classifier and describe the relative strengths/weakness of each classifier. Which classifier has overall best Performance?**

The confusion matrices of the USPS and MNIST data set can be referred from pages above.
The Relative strengths and weaknesses of each classifier that I found are:
Logistic Regression:

- Logistic Regression performed poorly on USPS data set, but it was better then the performance of Random Forest and SVM.
- Training time is greater for Logistic Regression

Random Forest:

- One of the disadvantages of Random Forest is that it easily overfits.
- Random Forest also performed very poorly on USPS data set.

Support Vector Machine:

- SVM doesn't work well with default parameters.
- SVM also performed very poorly on USPS data set.

Neural Network:

- Neural Network gives the best accuracy in a reasonable time.
- It also gives the highest accuracy on USPS data set among all the four classifiers.

- **Combine the Result of individual classifier using the classifier combination method such as majority voting. Is the overall combined performance better than that of any individual classifier?**

The ensemble of classifier is done using Majority Voting. The details are presented in the pages above. The accuracy obtained using ensemble classifier is shown below:
The accuracy of ensemble on MNIST test set = 0.9596
Accuracy of ensemble on USPS data set = 0.1837
The overall combined performance of ensemble classifier is better than any of the models on MNIST dataset but this is not true for USPS dataset, as Neural Network gives more accuracy for USPS data set.

References

- https://towardsdatascience.com/what-are-hyperparameters-and-how-to-tune-the-hyperparameters-in-a-deep-neural-network-d0604917584a
- https://machinelearningmastery.com/logistic-regression-for-machine-learning
- https://stats.stackexchange.com/questions/176209/combining-several-features-into-one-single-feature
- https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72