GROUP 18 Report(Assignment 3)

Department of Computer Science and Engineering
University at Buffalo, Buffalo, NY 14260

Hemanth Kumar Mutta **Desireddy Sai Sankeerthana** **Sai Sandeep Lankisetty**
hmutta@buffalo.edu **saisanke@buffali.edu** **slankise@buffalo.edu**
(50475286) **(50465523)** **(50469202)**

1.Problem 1: Logistic Regression:One Vs All

Implementation of logistic regression to classify hand -written digits into a correct corresponding label. This strategy uses concatenation of binary sigmoid functions. The label associated with each digit there would be 10 possible values for **blrObjFunction()** function as follows.

error for Train data:

Digit	Mean	Sum
0	0.0311	9.65
1	0.0283	9.39
2	0.0806	21.36
3	0.0953	23.83
4	0.0597	16.48
5	0.1009	23.12
6	0.0454	13.85

7	0.0536	14.84
8	0.1335	27.78
9	0.1210	25.78

Table1:train error data

error for Test data :

Digit	Mean	Sum
0	0.0138	4.904
1	0.0118	4.525
2	0.0709	18.31
3	0.0670	17.62
4	0.0420	12.31
5	0.0854	20.34
6	0.0288	9.15
7	0.0381	11.14
8	0.108	21.92
9	0.107	23.97

Table2:test error data

Observation:

As we see the above are the train and test error for function blrObj() from the above data we have mean and sum for train and test error. We got higher train error in mean and sum when compared to test error.

Difference between training error and test error:

Train Error: it is the error we get by calculating the classification error of model on same data from which model was trained.

Test Error: it is the error we get by using two completely disjoint datasets where one is from training the model and other from classification error .error on data which was unknown in training phase is test error(train error)

If test error is higher than train error that might lead to over-fitting. As we observe from above data we can say we that high train error when compared to test error in blrObj() function.

Overall Results for training, validation and testing accuracy for blr as follows:

Training Set	92.73%
Validation Set	91.5%
Testing Set	91.92%

Table 3:overall accuracy for blr

The above are the overall accuracies for training set, validation set and testing set in one Vs all strategy

Confusion Matrics for the test,train and validation set for blr as follows:

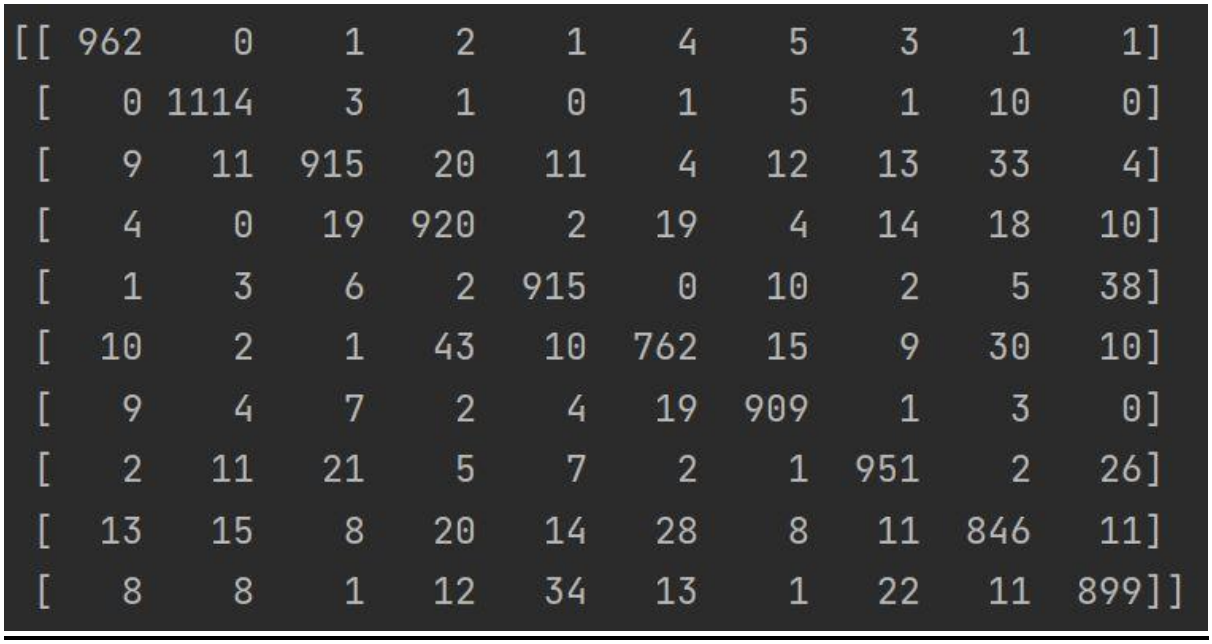

Fig1 :Confusion matrix for test set

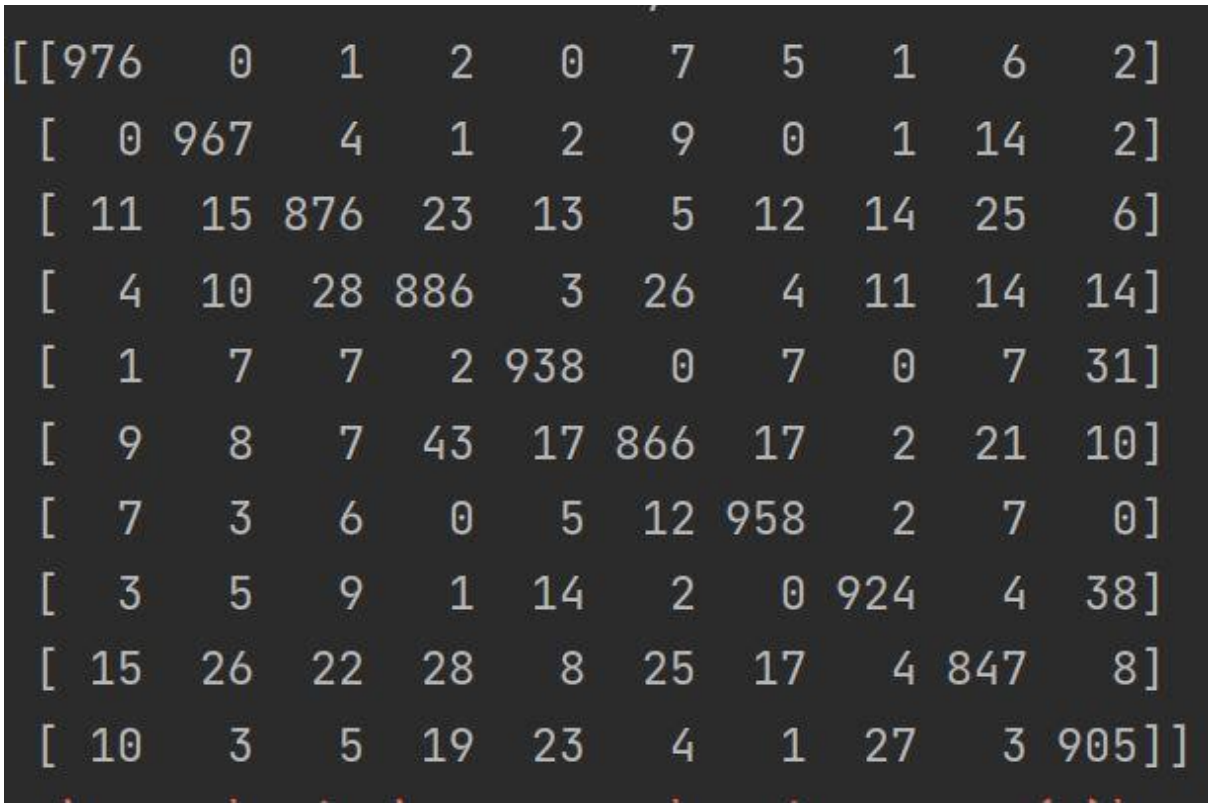

Fig2: Confusion matrix for validation set

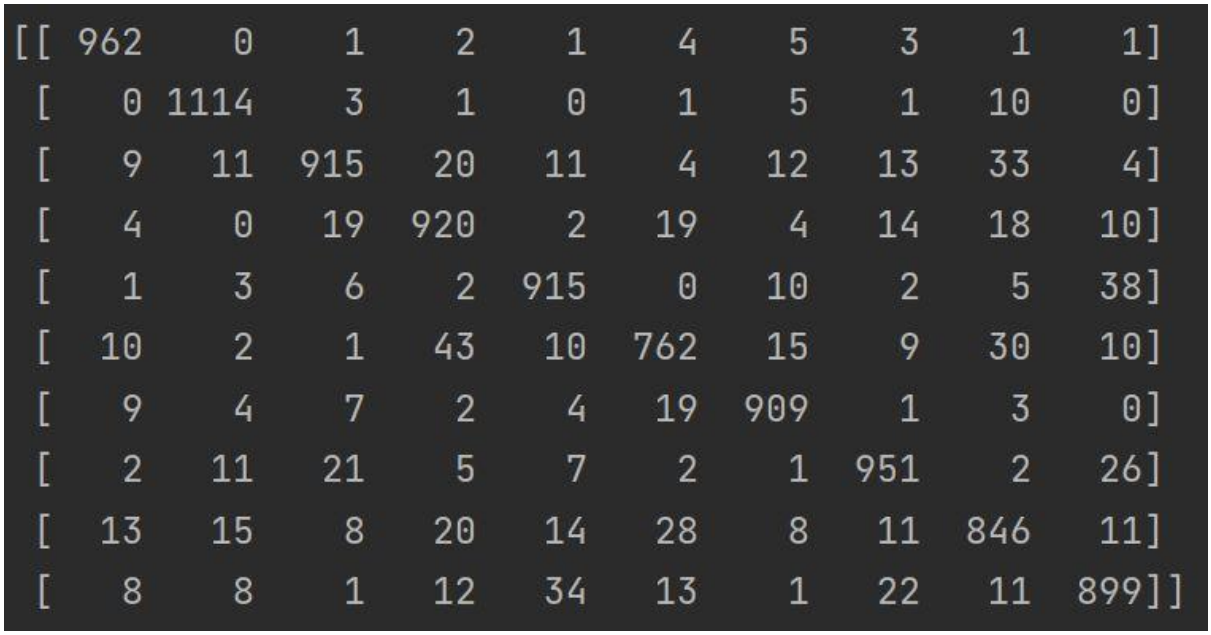

Fig 3: Confusion matrix for Training set.

Problem 1.2 Logistic Regression: Multi-class

Implementation of multi-class Logistic regression for classifier here we use only single classifier that is trained with softmax transformation of linear function method on the results of regressor. the following are results for train and test error for multi-class function.As the model is trained by using only one train set we can only view test set after completion of training.

Total error with respective (mean,sum) for training and test data as follows:

	Train Error	Test Error
Mean	0.3105	0.2222
Sum	59.93	43.10

Table 4:Error for test and train data

Overall Results for training, validation and testing accuracy as follows:

Training Set	93.485%
Validation Set	92.473%
Testing Set	92.556%

Table 5:overall accuracy for mlr

Confusion Matrics for the test,train and validation set for mlr as follows:

```
[[ 960    0    0    3    0    6    6    4    1    0]
 [   0 1110    3    2    0    2    4    2   12    0]
 [   6    8  924   16   10    3   14    8   39    4]
 [   4    1   20  914    0   25    3   10   26    7]
 [   1    1    6    2  921    0    9    4    9   29]
 [  10    2    2   37   10  773   15    6   30    7]
 [   9    3    4    2    7   15  914    3    1    0]
 [   1    9   19    6    6    2    0  952    2   31]
 [   9    8    6   26    9   23   10    8  868    7]
 [  11    8    0   10   28    5    0   20    8  919]]
```

Fig4:Confusion matrix for Test data

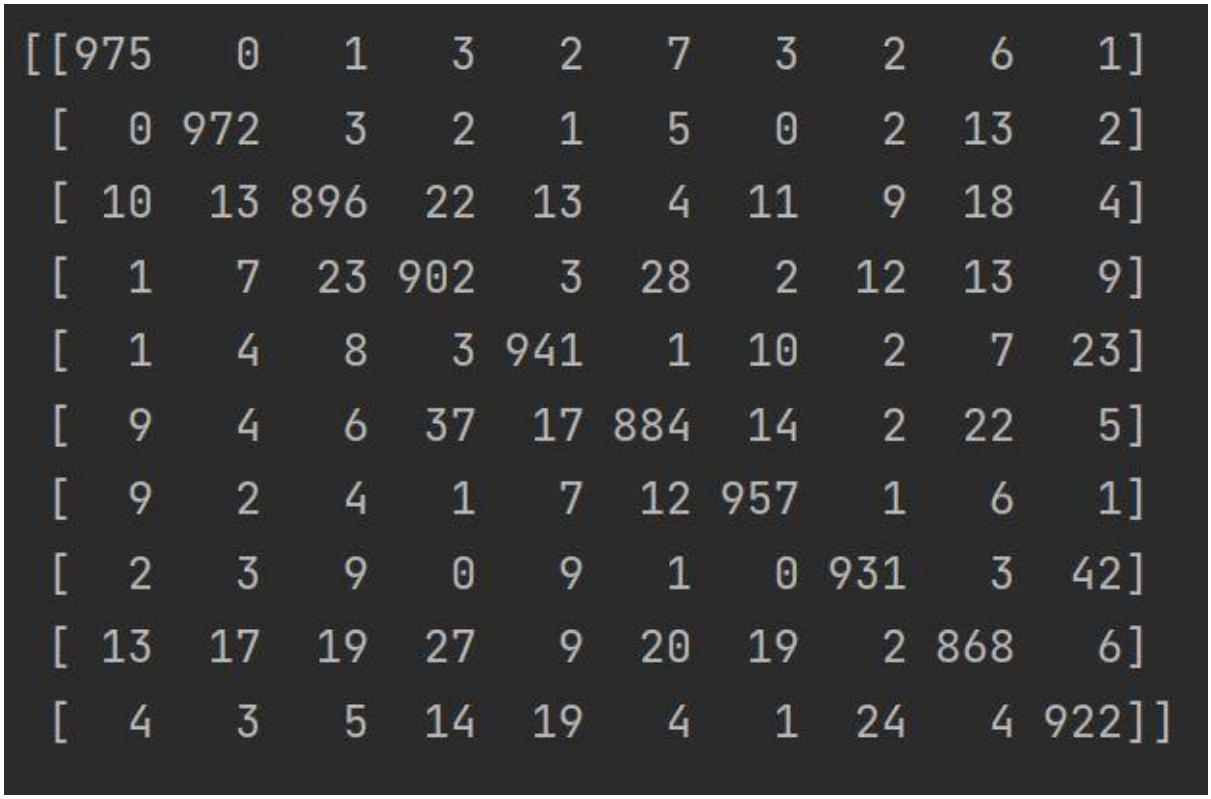

Fig5:Confusion Matrix for validation data

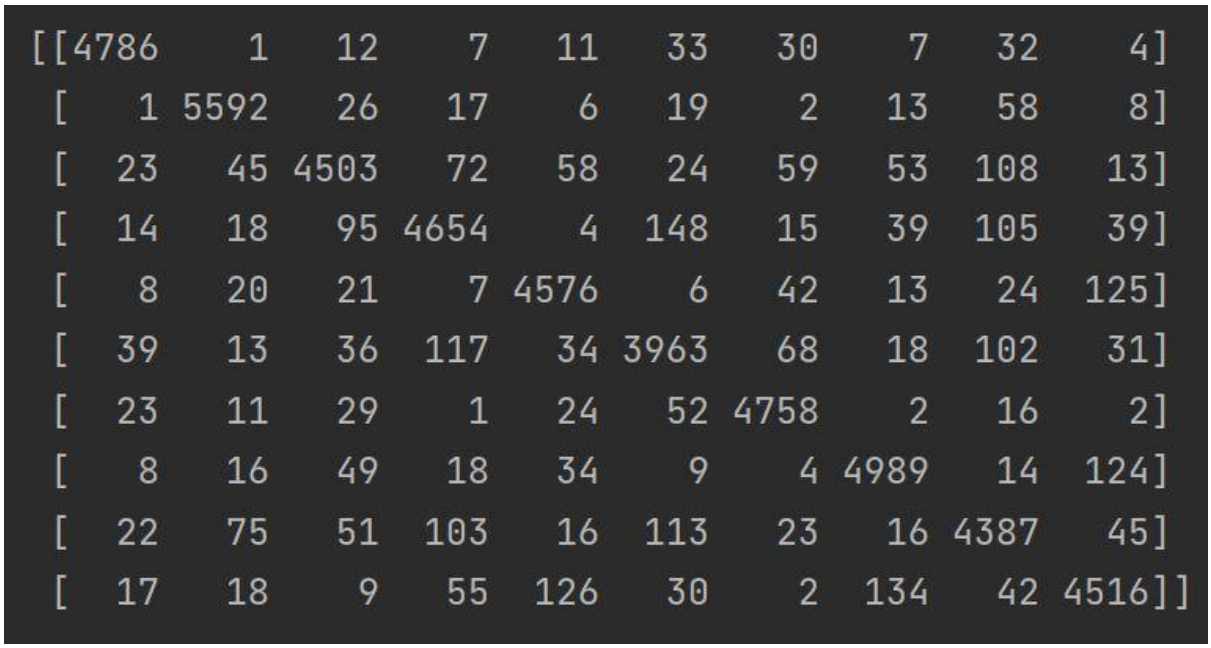

Fig 6:Confusion Matrix for train data

Difference between training error and test error:

As we observe the data in the table we can say that we have achieved higher train error than test error. We only have one value for train and test error because multi-class regressor was trained uniformly for all the classes.

Compare the performance difference between multi-class strategy with one-vs-all strategy:

From viewing all information above we can clearly say that Multi-Class Logistic regression has yield much better results when compared to One vs All strategy.

- In One Vs All strategy we need to train the scales with number of classes specified but in multi-class strategy we only need to train one scale that classify all the class at same time
- As Multi-class regressor requires only 1 classifier irrespective of number of classes . it trains the model much faster when compared to One vs All strategy.
- As we see the Mean/Sum of train and test errors have been increased in multi-class strategy when compared to One vs All strategy .
- The overall test set accuracy of one vs all strategy is 91.92% whereas overall test set accuracy for multi-class strategy is 92.556% by this we can also say multi-class strategy is best as it results better for test set data.

Problem 2:SVM(Support Vector Machine):

this model has been trained on small set of training data. Support Vector Machine(SVM) can be regularized by using gamma and C parameters. Any training sample influence is managed by gamma setting. training sample effect constrained to immediate vicinity by larger gamma values

the performance differences between linear kernel and radial basis, different gamma setting as follows:

Kernel	Gamma Setting	Training Accuracy	Validation Accuracy	Testing Accuracy
Linear	1	97.28 %	93.64%	93.78%
Radialbias function(rbf)	1	100%	15.47%	17.14%
Radialbias function(rbf)	0	98.98%	97.89%	97.87%

Table 6: overall accuracy for kernels

the above table is overall accuracy for train set, validation set and test set with respect to linear with default gamma setting as 1 and rbf with gamma 0,1

Observation:

The time taken to train in linear kernel is 5min37sec.time taken to train in rbf with gamma setting 0 is 5949sec.As we observe from the above table, Radial bias function when set with gamma value as 1 provides 100% accuracy for Training data .Which means it is performing better on training data when compared to linear kernel and rbf with gamma setting 0 but it performs poorly on validation and test that is due to over-fitting. From the above data, We can also say that radial bias function when set to gamma setting 1 has better results on test data which is 97.87% when compared to test data in linear kernel.

Radial bias function with varying C values as follows:

C	Training Accuracy	Validation Accuracy	Testing Accuracy
1	98.98%	97.89%	97.87%
10	99.98%	98.45%	98.34%
20	100%	98.44%	98.31%
30	100%	98.44%	98.31%
40	100%	98.44%	98.31%
50	100%	98.44%	98.31%
60	100%	98.44%	98.31%
70	100%	98.44%	98.31%
80	100%	98.44%	98.31%
90	100%	98.44%	98.31%
100	100%	98.44%	98.31%

Table 7: overall accuracy for varying c values

C is called as Penalty Factor in Support Vector Machine(SVM).it is a hyper-parameter in SVM to control error.. From the above listed data we can say that radial bias function-kernel with C value being 20 has performed best.

Plotting for Radial bias function for Varying C values

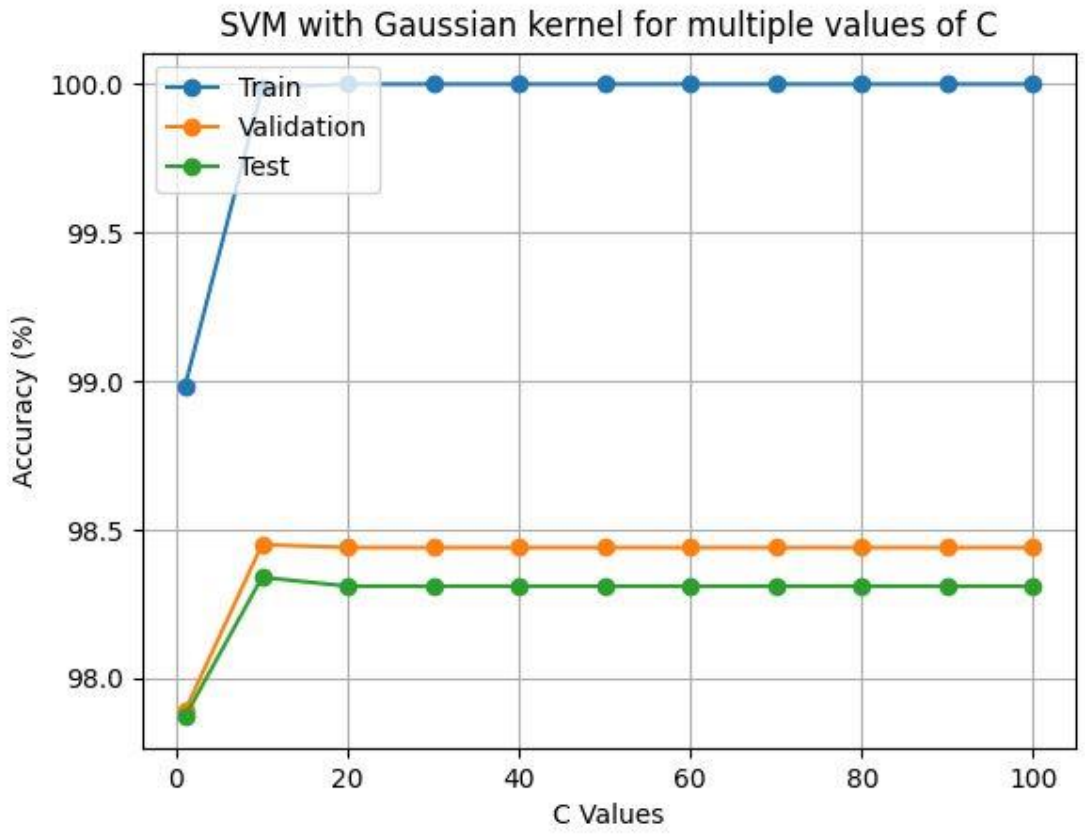

Fig 7:SVM with Gaussian Kernel for multiple C values

If you observe from the above graph, we can say that there exist a constant Training Accuracy, Testing Accuracy and validation Accuracy from C=20 to C=100 this might be due to overfiiting. Choosing C values for Kernel is a crucial job become choosing high values for C might lead to overfitting and choosing low values for C might lead to under-fitting , so values of C should be choosen wisely and carefully. Whereas,SVM as a disadvantage that is SVM is very prone to over-fitting .

CONCLUSION.

For the Problem MNIST digit , from above three methods we recommend multi-class logistic regression is best when compared to one vs all logistic regression and svm because in multi class logistic we need build only one classifer so trains data faster ,it is simple to use, yield much better results.