CSE4/574 Fall 2022 Introduction to Machine Learning
Programming Assignment 3

Classification and Regression

Team
Name: Mallikharjuna Rao Annam
UBIT Name : mallikha(50465177)
Name: Sai Sridhar Reddy Palli
UBIT Name : spalli (50468339)
Name : Kajol
UBIT Name : Kajol (50478691)

Binary Logistic Regression:

	Accuracy	Error
Training Set	92.667 %	7.333 %
Validation Set	91.46 %	8.54 %
Testing Set	91.94 %	8.06 %

From the above results of BLR of 3 parts of data (training,validation.testing). By comparing the values mentioned in the above table. We can say that the training data set is having a less error value while compared to the test data. So we can conclude that usually linear models work better on trained data, better when compared to test data but with a very minimal error change.

Following are the Training errors for each category

```
Class 1
0.020196602468822045
Class 2
0.02147816735089124
Class 3
0.062210526446993535
Class 4
0.07546634687911888
Class 5
0.04456000189012802
Class 6
0.0830280438747608
Class 7
0.034033179647582644
Class 8
0.043851216745196146
Class 9
0.11135068487285606
Class 10
0.0964572729939047
```

Following are the Testing errors for each category

```
Class 1
0.0013204558814456514
Class 2
0.0010941176621803038
Class 3
0.04511881625646233
Class 4
0.04131967780614742
Class 5
0.02130458333132088
Class 6
0.05625810839014929
Class 7
0.013595871181251926
Class 8
0.022270750716266045
Class 9
0.08257261821914923
Class 10
0.07636921097108562
```

Multi-class Logistic Regression:

	Accuracy	**Error**
Training Set	93.176	6.824 %
Validation Set	92.46	7.54 %
Test Set	92.51	7.49 %

From the above results of Multi class logistic regression of 3 parts of data (training,validation.testing). By comparing the values mentioned in the above table. We can say that the training data set is having a less error value while compared to the test data. So we can conclude that usually linear models work better on trained data, better when compared to test data but with a very minimal error change.

Difference in Performance between Binary Logistic Regression vs Multi-class Logistic Regression:

	BLR Accuracy	MLR Accuracy
Training Set	92.667	93.176
Validation Set	91.46	92.46
Test Set	91.94	92.51

Since the major difference between the Binary Logistic Regression and Multi Class Logistic Regression is, BLR works in a way that i can't classify more than a single class so it considers one class at a time, it requires 10 BLR's to classify for 10 classes. while in MCLR it classifies all the classes at a single time.

Support Vector Machine:

i) Using Linear Kernel:

	Accuracy
Training Set	92.784 %
Validation Set	91.869 %
Testing Set	91.77 %

From the above results of Support vector machine with linear kernel, of 3 parts of data (training,validation.testing). By comparing the values mentioned in the above table. We can say that the training data set is having a less error value while compared to the test data. So we can conclude that since we are using the linear kernel,it usually works like linear models work better on trained data, better when compared to test data but with a very minimal error change.

Radial Basis Function:

i) Using Radial Basis Function(Gamma Value = 1)

	Accuracy
Training Set	100.0 %
Validation Set	10.0 %
Testing Set	11.35 %

By seeing the training dataset accuracy we can conclude that the model is overfitting. Since we changed the kernel from linear to radial basis function of gamma value ‘1’. This will obviously give very low accuracy on test data. Because the model is able to give high accuracy on the training data.

ii) Using Radial Basis Function (Gamma Value = default)

	Accuracy
Training Set	92.07 %
Validation Set	91.85 %
Testing Set	92.369 %

Now instead of changing the gamma value, if we use the default settings of the radial basis function, then compare the above respective values of training, testing, validation datasets. It shows good accuracies on both testing and training data. In fact test data is getting the little more accuracy than training data.

C	Training Accuracy	Validation Accuracy	Test Accuracy
1	96.468	95.99	95.04
10	97.124	96.81	95.83
20	97.158	96.69	96.67
30	97.158	96.69	96.67
40	97.158	96.69	96.67
50	97.158	96.69	96.67
60	97.158	96.69	96.67
70	97.158	96.69	96.67
80	97.158	96.69	96.67
90	97.158	96.69	96.67
100	97.158	96.69	96.67

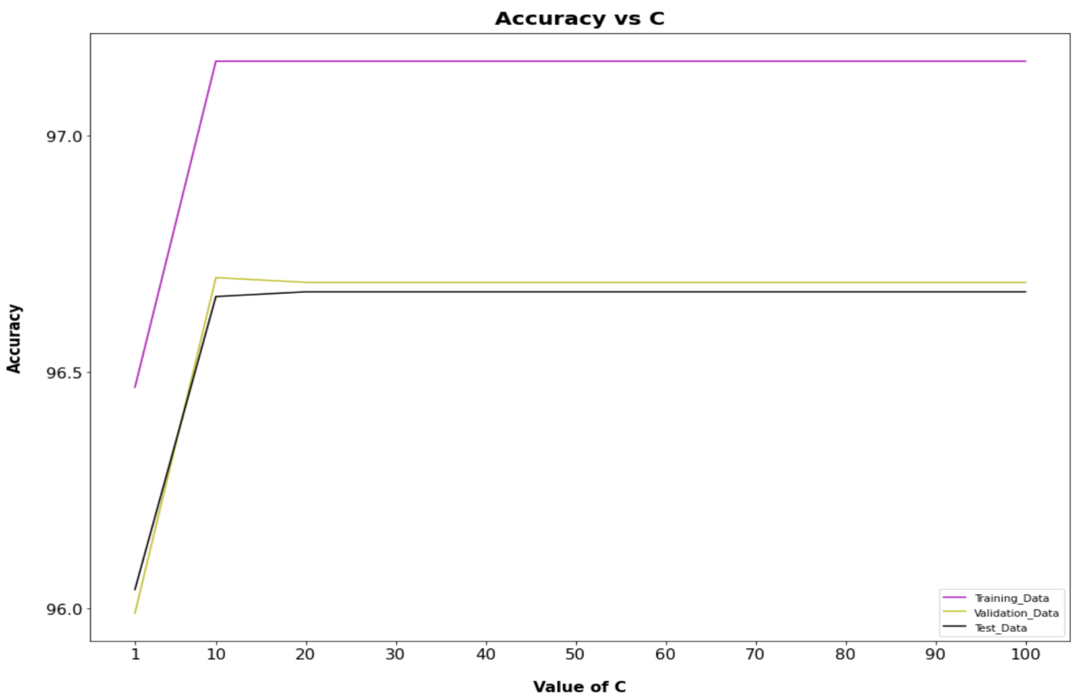
In the above when we used the Gamma value ‘1’ the training data shows 100 % accuracy, Now by iterating the values from [1,10,20,30,...100] of ‘c’. By seeing the values from the above table there is a slight increase in the accuracy when the ‘c’ value is getting increased.

Best Iteration :

Kernel	C	Training Accuracy	Validation Accuracy	Testing Accuracy
RBF (Gamma Value - default)	20	99.339	97.36	97.26

From the above iteration on multiple variations of ‘c’ values using the radial basis we can conclude that we are getting the best result by setting gamma to default and taking C = 20

Following is the graph, obtained by plotting the different values of ‘c’ for each training,testing and validation dataset.



Observation:

When the dataset is vast, multidimensional and high with information then Linear Kernel is best. But, the MNIST has dimensions but it doesn't actually contain much information. So, the accuracy is not as high as compared to non-linear models.

Low accuracy is observed when gamma = 1 compared to gamma = 0. Increasing the value of Penalty factor C value also increases the accuracy.

In Regularization, if proper C value is not chosen then it is possible to store many support vectors that either underfit or overfit the data.If we opt for smaller or the higher C value respectively. By seeing all these we can conclude that choosing proper value for C is important.

The SVM will classify most of the training data points correctly if the value of C is large. But for smaller values of C it misclassifies.