DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY AT BUFFALO

CSE 574 Programming Assignment 3
Classification And Regression

Group 141

Shivam Sahu
(#50247673)

Parush Garg
(#50248921)

PART 1 - Implementation of Logistic Regression:

Data	Accuracy (%)
Training data	84.896
Validation data	83.74
Test data	84.08

Time taken for Logistic Regression - 1488.4046782690566 secs (Approx. 24 min)

Training set Confusion Matrix:

```
[[4833    1   14    8   10   20   25    6    2    4]
 [   2 5655   33   13    3   17    3   10    0    6]
 [  36   41 4588   76   50   28   55   66    1   17]
 [  21   25  133 4665    9  143   21   46    1   67]
 [   8   21   24    5 4573   13   24   14    0  160]
 [  51   19   38  135   47 3953   93   17    8   60]
 [  27   13   30    2   22   70 4745    3    3    3]
 [  12   22   48   11   46   13    3 4974    0  136]
 [ 131  292  841 1005  192 1310  131   52   35  862]
 [  29   23   13   88  166   44    1  156    2 4427]]
```

```
Error for class 1 : 1.82815 %
Error for class 2 : 1.51515 %
Error for class 3 : 7.46269 %
Error for class 4 : 9.08205 %
Error for class 5 : 5.55556 %
Error for class 6 : 10.5858 %
Error for class 7 : 3.51769 %
Error for class 8 : 5.52707 %
Error for class 9 : 99.2785 %
Error for class 10 : 10.5476 %
```

Testing set Confusion Matrix:

```
[[ 961    0    1    3    1    5    5    3    0    1]
 [   0 1123    4    1    0    2    4    1    0    0]
 [   9   11  942   22   12    4   14   14    0    4]
 [   4    1   22  928    2   22    4   14    0   13]
 [   1    2    6    3  917    0    9    2    1   41]
 [  11    4    3   42   14  777   21    9    0   11]
 [   9    4    8    2    4   20  910    1    0    0]
 [   2    9   22    7    7    2    1  948    1   29]
 [  33   41  150  203   51  258   47   19    1  171]
 [  10    8    1   16   35   15    1   22    0  901]]
```

```
Error for class 1 : 1.93878 %
Error for class 2 : 1.05727 %
Error for class 3 : 8.72093 %
Error for class 4 : 8.11881 %
Error for class 5 : 6.61914 %
Error for class 6 : 12.8924 %
Error for class 7 : 5.01044 %
Error for class 8 : 7.7821 %
Error for class 9 : 99.8973 %
Error for class 10 : 10.7037 %
```

Conclusion:-

The accuracy given by Logistic Regression is less because Logistic Regression considers all the points in a dataset and choose any hyperplane which separates the data rather than choosing the best hyperplane.

PART 2: Implementation Of Support Vector Machines

Activity 1 - SVM with linear kernel

Kernel	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Time (in secs)
linear	97.286	93.64	93.78	950.766103

Time taken for SVM with linear kernel :— 950.766103 secs (Approx. 15.8 min)

Activity 2 - SVM with radial basis function, gamma = 1

Kernel	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Time (in secs)
rbf	100	15.48	17.14	16157.1364

Time taken for SVM with radial basis function, gamma 1 :— 16157.136420 secs (Approx. 4.5 hours)

Activity 3 - SVM with radial basis function, gamma = 0

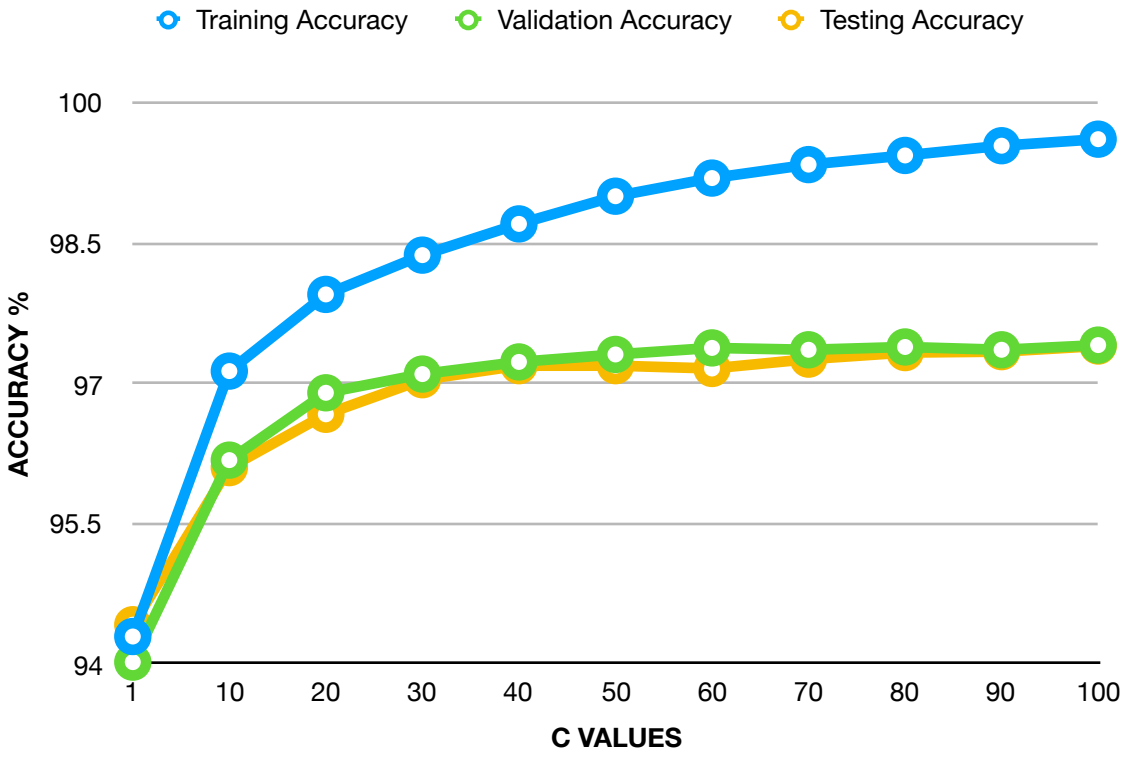
Kernel	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Time (in secs)
rbf	94.294	94.02	94.42	1677.707449

Time taken for SVM with radial basis function, gamma 0 :— 1677.707449 secs (Approx. 28 min)

Activity 4 - SVM with radial basis function, different values of C

Kernel	C Values	Training Accuracy	Validation Accuracy	Testing Accuracy
rbf	1	94.294	94.02	94.42
rbf	10	97.132	96.18	96.1
rbf	20	97.952	96.9	96.67
rbf	30	98.372	97.1	97.04
rbf	40	98.706	97.23	97.19
rbf	50	99.002	97.31	97.19
rbf	60	99.196	97.38	97.16
rbf	70	99.34	97.36	97.26
rbf	80	99.438	97.39	97.33
rbf	90	99.542	97.36	97.34
rbf	100	99.612	97.41	97.4

Time taken for SVM with radial basis function, different values of C :— 10214.1234 secs (Approx. 2.83 hours)



Observations:-

1. **Activity 1 :** Linear kernel is useful when the dataset have lots of features i.e multi-dimensional data and where original data is very informative. Our MNIST dataset have high dimensionally but the pixels are not very informative hence the accuracy is not as high as it would be for nonlinear models.

2. Low accuracy is observed in **Activity 2** where gamma = 1 compared to **Activity 3** where gamma = 0 which shows a clear case of overfitting.

3. From **Activity 4** We can observe the below points:

a. From graph we can see that with increase in C, accuracy also increases.

b. In any regularization scheme, it is important to choose proper value for C, also known as the penalty factor. If it is too large, we have a high penalty for non-separable points and we may store many support vectors and overfit. If it is too small, we may have under-fitting.

c. For large values of C like 50 and above, the SVM will choose a smaller-margin hyperplane which classifies more of training data points correctly.

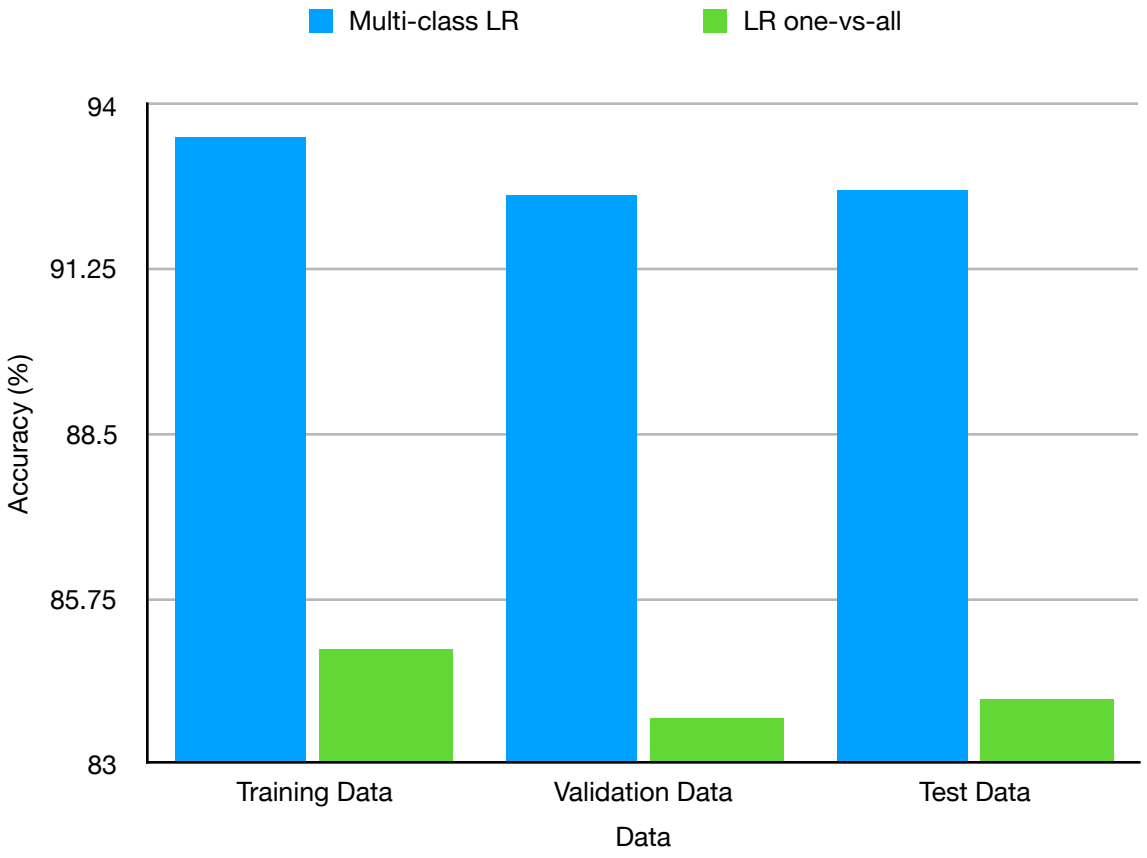
d. Conversely, a very small value of C like less that 50, will cause the optimizer to choose a larger-margin separating hyperplane, even if it misclassifies more data points.

Conclusion:-

So from **Part 1** and **Part 2** we can conclude that SVM (with rbf kernel) works better when the dataset have lots of features i.e multi-dimensional data and Logistic Regression works better when dataset have fewer feature. Our MNIST dataset have more no of features thats why SVM gives better accuracy than Logistic Regression

PART 3 - Implementation of Multi-class Logistic Regression

Method	Training Data	Validation Data	Test data	Time (in secs)
Multi-class Logistic Regression	93.448	92.48	92.55	88.505324
Logistic Regression one-vs-all	84.896	83.74	84.08	1488.40467



Training set Confusion Matrix:

```
[[4786    1   12    7   11   33   30    7   32    4]
 [   1 5592   26   17    6   19    2   13   58    8]
 [  23   45 4503   72   58   24   59   53  108   13]
 [  14   18   95 4654    4  148   15   39  105   39]
 [   8   20   21    7 4576    6   42   13   24  125]
 [  39   13   36  117   34 3963   68   18  102   31]
 [  23   11   29    1   24   52 4758    2   16    2]
 [   8   16   49   18   34    9    4 4989   14  124]
 [  22   75   51  103   16  113   23   16 4387   45]
 [  17   18    9   55  126   30    2  134   42 4516]]
```

```
Error for class 1 : 2.78286 %
Error for class 2 : 2.61233 %
Error for class 3 : 9.17709 %
Error for class 4 : 9.29644 %
Error for class 5 : 5.4936 %
Error for class 6 : 10.3596 %
Error for class 7 : 3.25335 %
Error for class 8 : 5.24216 %
Error for class 9 : 9.56504 %
Error for class 10 : 8.74924 %
```

Testing set Confusion Matrix:

```
[[ 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]]
```

```
Error for class 1 : 2.04082 %
Error for class 2 : 2.20264 %
Error for class 3 : 10.4651 %
Error for class 4 : 9.50495 %
Error for class 5 : 6.21181 %
Error for class 6 : 13.3408 %
Error for class 7 : 4.5929 %
Error for class 8 : 7.393 %
Error for class 9 : 10.883 %
Error for class 10 : 8.91972 %
```

Conclusion:-

Logistic Regression traditionally is used for binary classification. Multi-class Logistic Regression can be used to build a classifier that can classify 10 classes at the same time. In term of performance from the table above we are able to observe that the accuracy obtained in Multi-class Logistic Regression is higher comparing to one-vs-all Logistic Regression strategy. Moreover, since we are able to classify 10 classes simultaneously, the time to complete for Multi-class Logistic Regression is less.