CSE574 Introduction to Machine Learning
Programming Assignment 4
Classification and Regression

Submitted by
1. Sahithya Arveti Nagaraju - 50559752
2. Sushmitha Manjunatha – 50560530

Binary Logistic Regression(BLR):

Logistic regression is a statistical method used to estimate or predict outcomes based on a set of independent variables, specifically when the dependent variable is binary (takes only two values). For binary logistic regression applied to the MNIST dataset, the following results were obtained:

Set	Accuracy	Error
Training	92.65%	7.35%
Validation	91.49%	8.51%
Testing	91.97%	8.03%

The results above represent the performance of Binary Logistic Regression on the Training, Validation, and Testing datasets. It is evident that the training error is lower than the testing error, indicating that the model performs better on the data it has seen during training. However, when applied to unseen data, the error increases slightly. This behavior is typical for linear models.

Reason for the Difference:

- **Overfitting:** Although minimal, the model might slightly overfit the training data, resulting in a marginally better performance on it.
- **Data Similarity:** The small error difference suggests that the training, validation, and test sets share similar distributions, making generalization easier for the logistic regression model.

Multi-class Logistic Regression (MLR):

Set	Accuracy	Error
Training	93.44%	6.56%
Validation	92.47%	7.53%
Testing	92.55%	7.45%

The results above show the performance of Multi-class Logistic Regression on the Training, Validation, and Testing datasets. The training error is slightly lower than the testing error, indicating that the model performs better on the data it has seen during training. However, when evaluated on unseen data, the error increases marginally. This behavior is typical for linear models. The near equality of errors across the datasets suggests that the data distribution and patterns are consistent across the training, validation, and testing sets, leading to comparable model performance.

Performance difference between multi-class strategy(MLR) with one-vs-all(BLR) strategy:

Set	MLR Accuracy	BLR Accuracy
Training	93.44%	92.65%
Validation	92.47%	91.49%
Testing	92.55%	91.97%

In multiclass logistic regression, all 10 classes of the MNIST dataset are classified simultaneously, whereas in one-vs-all binary logistic regression (BLR), each class is classified against all others one at a time. This makes multiclass logistic regression more efficient, with lower time complexity and reduced chances of overlapping classifications.

Accuracy:

- MLR achieves higher accuracy across all datasets due to the collective estimation of parameters, which better leverages inter-class relationships and reduces classification overlap.
- BLR estimates parameters independently for each binary classification problem, increasing the risk of inconsistent predictions between classifiers.

Reasons for Error Differences:

- **Overfitting:** MLR slightly overfits the training data, resulting in a lower training error compared to testing error.
- **Data Distribution Consistency:** The similar error rates across datasets reflect a uniform distribution of patterns, reducing discrepancies in performance.
- **Model Simplicity:** Logistic regression's linear nature may struggle to capture non-linear patterns in MNIST data, capping its performance.

Support Vector Machine (SVM):

1. **Using Linear Kernel:**

Set	Accuracy
Training	97.00%
Validation	94.00%
Testing	94.00%

The Linear Kernel SVM achieves a commendable accuracy of 94%. This indicates that the Linear Kernel functions similarly to a linear model, as the results are closely aligned with those obtained from the previous linear model we trained.
This is expected, as the Linear Kernel SVM performs well on data with a linear decision boundary or data that can be approximated by linear separability. The performance across training, validation, and testing datasets is fairly consistent, indicating that the model is not overfitting and generalizes well to unseen data.

2. **Radial Basis Function:**

(a). Using Radial Basis Function (Gamma = 1)

Set	Accuracy
Training	100.0%
Validation	15.00%
Testing	17.00%

With Gamma = 1, the SVM using the RBF kernel achieves a perfect 100% training accuracy, but the validation and testing accuracy are significantly lower. This behavior is indicative of overfitting, as the model has memorized the training dataset and struggles to generalize to unseen data. A high Gamma value causes the model to be heavily influenced by individual training examples, reducing its ability to handle variations in the data effectively.

(b). Using Radial Basis Function with value of gamma setting to default (all other parameters kept as default)

Set	Accuracy
Training	99.00%
Validation	98.00%
Testing	98.00%

When Gamma is set to its default value, the SVM performs much better in terms of generalization. The model still achieves near-perfect accuracy on the training set (99%) but maintains a high level of accuracy on the validation and testing datasets (98%). This indicates that the SVM with the default Gamma value is more balanced, capable of capturing complex relationships in the data without overfitting. The slightly lower performance on the testing data compared to training data suggests that while the model is highly effective, there might still be a small degree of overfitting, though it's much less pronounced than with Gamma = 1.

(c) . Using Radial Basis Function with value of gamma setting to default and varying value of C (1, 10, 20, 30, …,100)

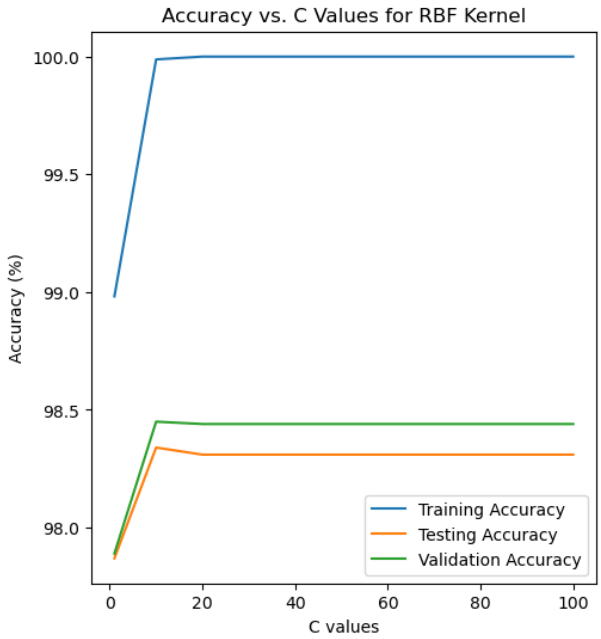
Next, we iterate through different C values to identify the optimal setting. Once determined, the entire dataset is tested using this configuration. The C parameter controls the trade-off between maximizing the margin and minimizing the slack variable, effectively balancing model complexity and misclassification. Here, we observe a trade-off between the margin's width and the C value.
Here are the results for different values of C on the Training, Validation, and Testing datasets:

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%

- As **C** increases, the model becomes more prone to overfitting, as seen in the perfect training accuracy (100%) with little improvement in validation or testing accuracy after **C = 20**. However, there is still no significant drop in testing accuracy even as **C** increases, indicating that after a certain point, the model stabilizes and does not gain further benefits in terms of generalization.
- The validation and testing accuracy remain relatively stable around 98%, demonstrating that the SVM is effectively balancing the model complexity and the error tolerance with **C = 10** to **C = 100**.

The plot below illustrates the accuracy achieved on the Training, Validation, and Testing datasets for various values of C:



This suggests that our dataset exhibits non-linear characteristics, as the non-linear model delivers better results compared to linear models