CSE574 Introduction to Machine Learning Programming Assignment 4
Classification and Regression

Nandhakumar Vadivel (50604851)
Sai Kalyan Veera (50560096)

Problem -1

Training set Accuracy:92.728%

Validation set Accuracy:91.53%

Testing set Accuracy:91.93%

Per-Class Training Errors:

Class 0: 2.19%

Class 1: 1.95%

Class 2: 8.81%

Class 3: 10.29%

Class 4: 6.05%

Class 5: 12.08%

Class 6: 3.74%

Class 7: 5.77%

Class 8: 12.62%

Class 9: 10.69%

Per-Class Testing Errors:

Class 0: 2.04%

Class 1: 1.76%

Class 2: 11.24%

Class 3: 9.01%

Class 4: 6.62%

Class 5: 14.57%

Class 6: 5.11%

Class 7: 7.68%

Class 8: 13.14%

Class 9: 11.00%

Total Training Error: 7.28%

Total Testing Error: 8.09%

Class 1 shows the lowest training and testing errors (1.95% and 1.76%, respectively), indicating that it is the easiest to classify.

Class 5 and Class 8 exhibit the highest errors in both training and testing datasets, suggesting that these classes are more challenging for the model to distinguish.

Difference Between Training and Testing Errors:

- The testing errors are consistently higher than the training errors for all classes. This is expected as the model might slightly overfit the training data, leading to reduced generalization performance on unseen test data.

The total training error (7.28%) is slightly lower than the total testing error (8.09%), indicating good but not perfect generalization.

<u>For Extra Credit</u>

1. Introduction

This report evaluates the performance of logistic regression models for the MNIST dataset using two approaches:

1. One-vs-All Logistic Regression: Builds 10 binary classifiers, one for each digit.
2. Multi-class Logistic Regression: Uses a single unified model to classify all 10 digits simultaneously.

The models are evaluated based on their total and per-class errors for both training and testing datasets. Insights into differences between training and testing errors are analyzed.

--------------Logistic Regression (One-vs-All)-------------------

Training set Accuracy:92.684%

Validation set Accuracy:91.49000000000001%

Testing set Accuracy:92.04%

Per-Class Training Accuracies and Errors:
Digit 0: Accuracy = 97.85%, Error = 0.1225

Digit 1: Accuracy = 97.88%, Error = 0.1075
Digit 2: Accuracy = 91.21%, Error = 0.4030
Digit 3: Accuracy = 89.59%, Error = 0.4906
Digit 4: Accuracy = 93.82%, Error = 0.2738
Digit 5: Accuracy = 88.26%, Error = 0.5877
Digit 6: Accuracy = 96.26%, Error = 0.2052
Digit 7: Accuracy = 94.21%, Error = 0.2479
Digit 8: Accuracy = 87.53%, Error = 0.7113
Digit 9: Accuracy = 88.93%, Error = 0.5763

Per-Class Test Accuracies and Errors:
Digit 0: Accuracy = 97.96%, Error = 0.1234
Digit 1: Accuracy = 98.24%, Error = 0.1051
Digit 2: Accuracy = 89.15%, Error = 0.5243
Digit 3: Accuracy = 90.89%, Error = 0.4197
Digit 4: Accuracy = 93.38%, Error = 0.2906
Digit 5: Accuracy = 85.87%, Error = 0.6151
Digit 6: Accuracy = 94.99%, Error = 0.2266
Digit 7: Accuracy = 92.51%, Error = 0.3471
Digit 8: Accuracy = 86.96%, Error = 0.6948
Digit 9: Accuracy = 89.20%, Error = 0.6029

--------------Logistic Regression (Multi-Class)-------------------

Training set Accuracy:93.33%

Validation set Accuracy:92.44%

Testing set Accuracy:92.58%

Per-Class Training Accuracies and Errors (Multi-Class):
Digit 0: Accuracy = 97.14%, Error = 0.1098
Digit 1: Accuracy = 97.56%, Error = 0.1094
Digit 2: Accuracy = 90.98%, Error = 0.3278
Digit 3: Accuracy = 90.76%, Error = 0.3144
Digit 4: Accuracy = 94.05%, Error = 0.2203
Digit 5: Accuracy = 89.82%, Error = 0.3759
Digit 6: Accuracy = 96.48%, Error = 0.1259
Digit 7: Accuracy = 94.32%, Error = 0.2088

Digit 8: Accuracy = 89.63%, Error = 0.3356
Digit 9: Accuracy = 91.51%, Error = 0.3114

Per-Class Test Accuracies and Errors (Multi-Class):
Digit 0: Accuracy = 97.86%, Error = 0.0788
Digit 1: Accuracy = 97.80%, Error = 0.1077
Digit 2: Accuracy = 90.21%, Error = 0.3888
Digit 3: Accuracy = 90.99%, Error = 0.2796
Digit 4: Accuracy = 93.48%, Error = 0.2200
Digit 5: Accuracy = 86.66%, Error = 0.4166
Digit 6: Accuracy = 95.30%, Error = 0.1729
Digit 7: Accuracy = 92.22%, Error = 0.2975
Digit 8: Accuracy = 88.60%, Error = 0.3918
Digit 9: Accuracy = 91.58%, Error = 0.3287

One-vs-All Logistic Regression Performance

Digit	Training Accuracy (%)	Training Error	Test Accuracy (%)	Test Error
0	97.85	0.1225	97.96	0.1234
1	97.88	0.1075	98.24	0.1051
2	91.21	0.4030	89.15	0.5243
3	89.59	0.4906	90.89	0.4197
4	93.82	0.2738	93.38	0.2906
5	88.26	0.5877	85.87	0.6151
6	96.26	0.2052	94.99	0.2266
7	94.21	0.2479	92.51	0.3471
8	87.53	0.7113	86.96	0.6948
9	88.93	0.5763	89.20	0.6029

Multi-Class Logistic Regression Performance

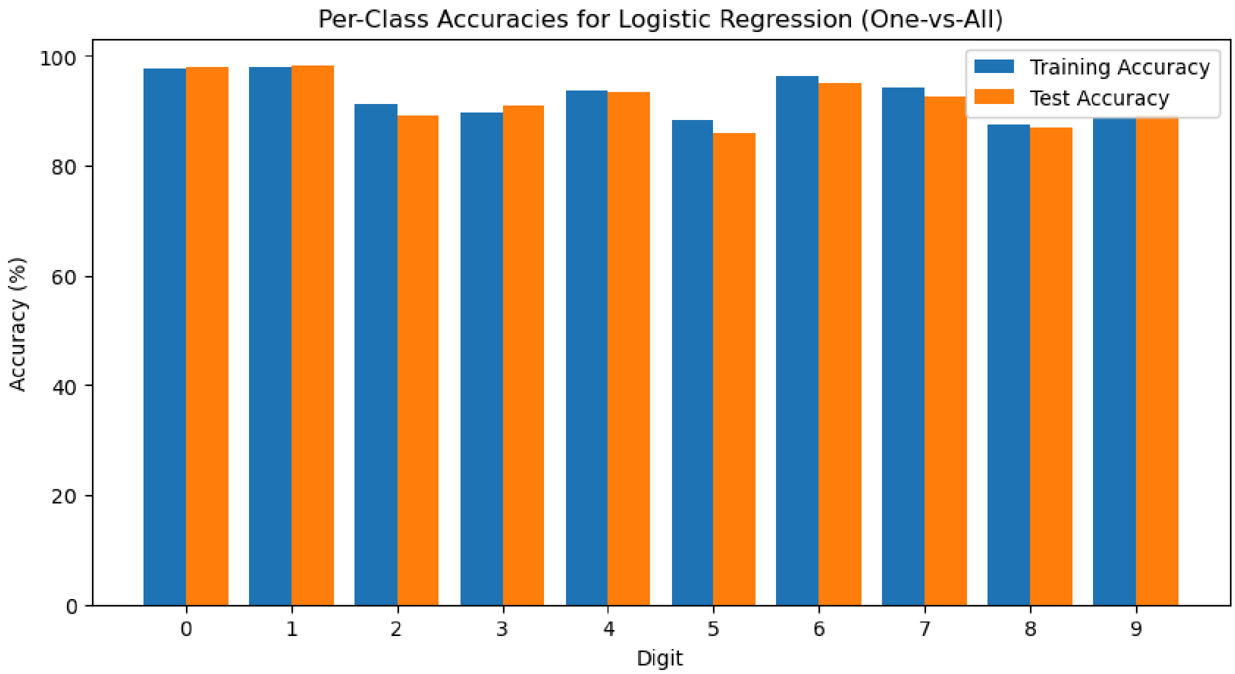
Digit	Training Accuracy (%)	Training Error	Test Accuracy (%)	Test Error
0	97.14	0.1098	97.86	0.0788
1	97.56	0.1094	97.80	0.1077
2	90.98	0.3278	90.21	0.3888
3	90.76	0.3144	90.99	0.2796
4	94.05	0.2203	93.48	0.2200
5	89.82	0.3759	86.66	0.4166
6	96.48	0.1259	95.30	0.1729
7	94.32	0.2088	92.22	0.2975
8	89.63	0.3356	88.60	0.3918
9	91.51	0.3114	91.58	0.3287

One-vs-All Logistic Regression:

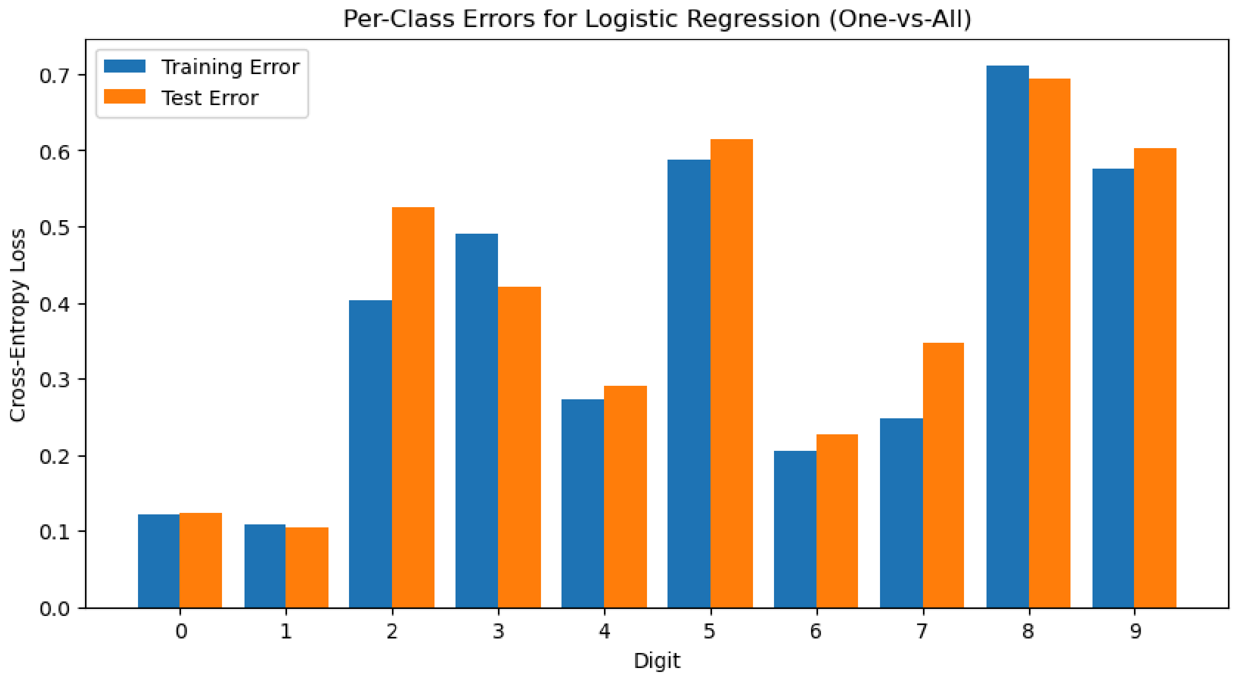
- Training Set Accuracy: 92.684%
- Validation Set Accuracy: 91.49%
- Testing Set Accuracy: 92.04%

Multi-Class Logistic Regression:

- Training Set Accuracy: 93.33%
- Validation Set Accuracy: 92.44%
- Testing Set Accuracy: 92.58%

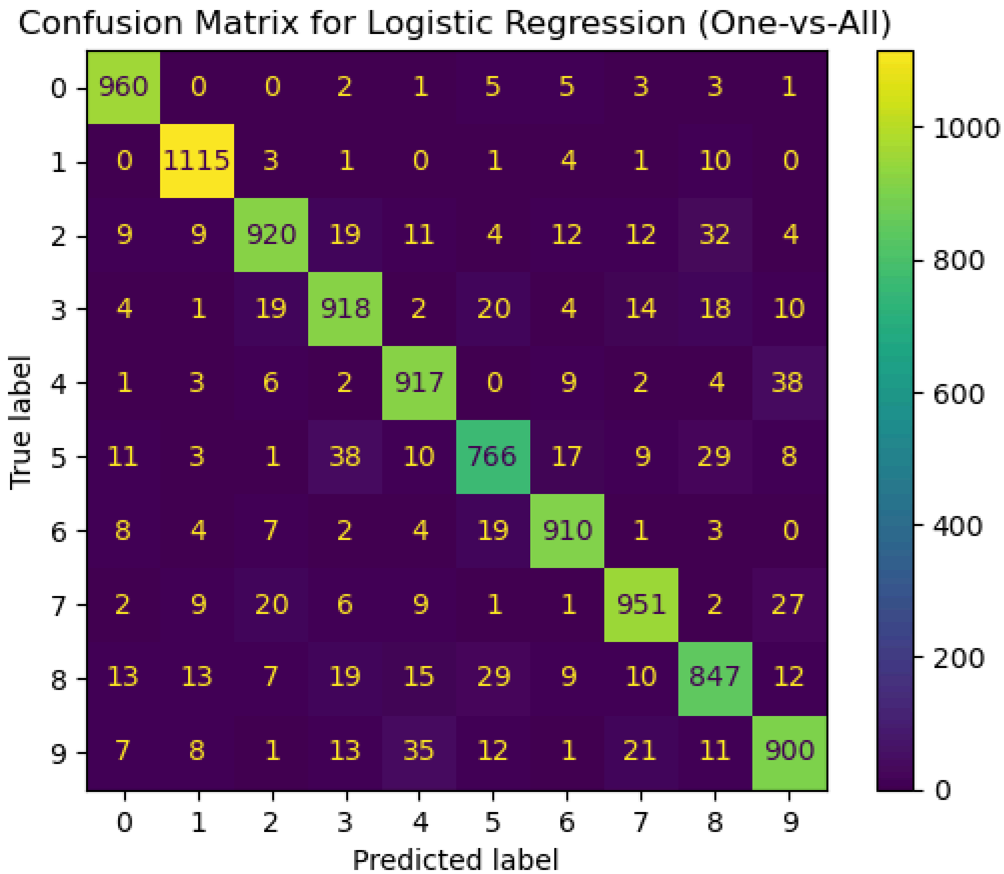


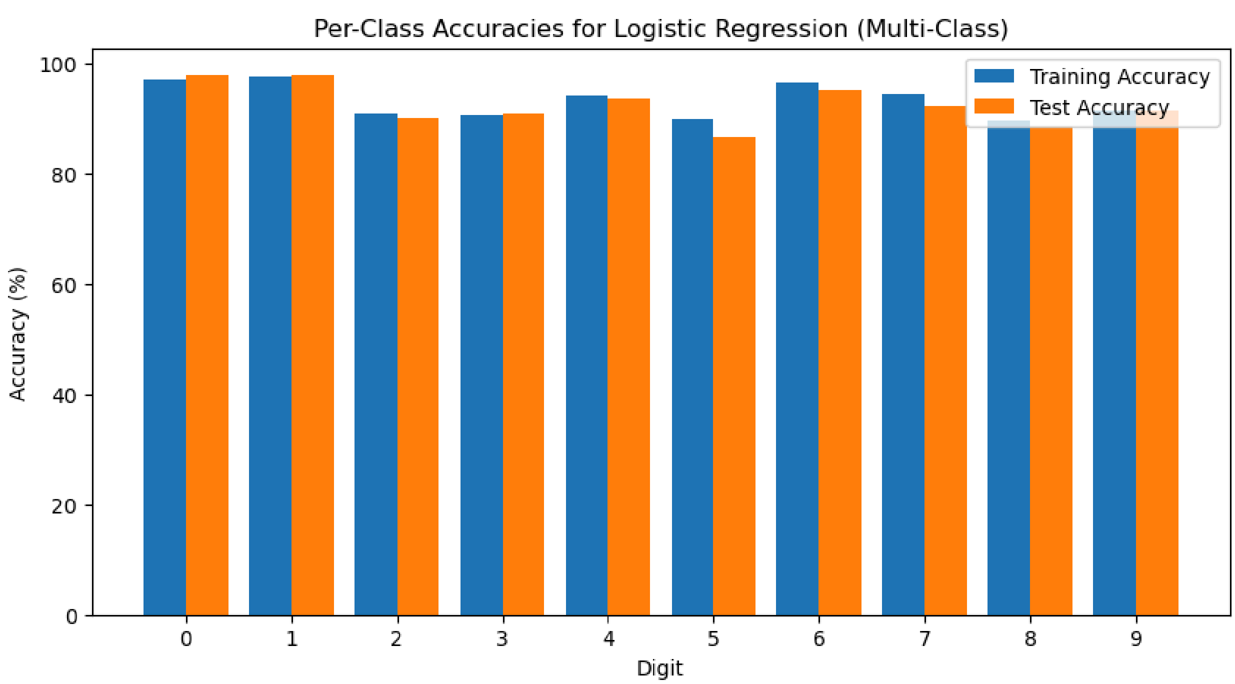
- Digits with High Accuracy: Digits 0, 1, 6, and 4 consistently show high accuracies (>93%) across both strategies and datasets.
- Digits with Lower Accuracy:
    - Digit 5: Exhibits lower accuracies and higher errors in both strategies. This may be due to its complex shape and similarity to other digits like 3 or 6.
    - Digit 8: Also shows lower performance, possibly because of its similarity to 0 or 3, and variations in handwriting styles.
- Comparison Between Strategies:
    - The multi-class logistic regression generally shows slightly higher training and test accuracies compared to the one-vs-all strategy.
    - The errors are also slightly lower in the multi-class strategy for some digits.

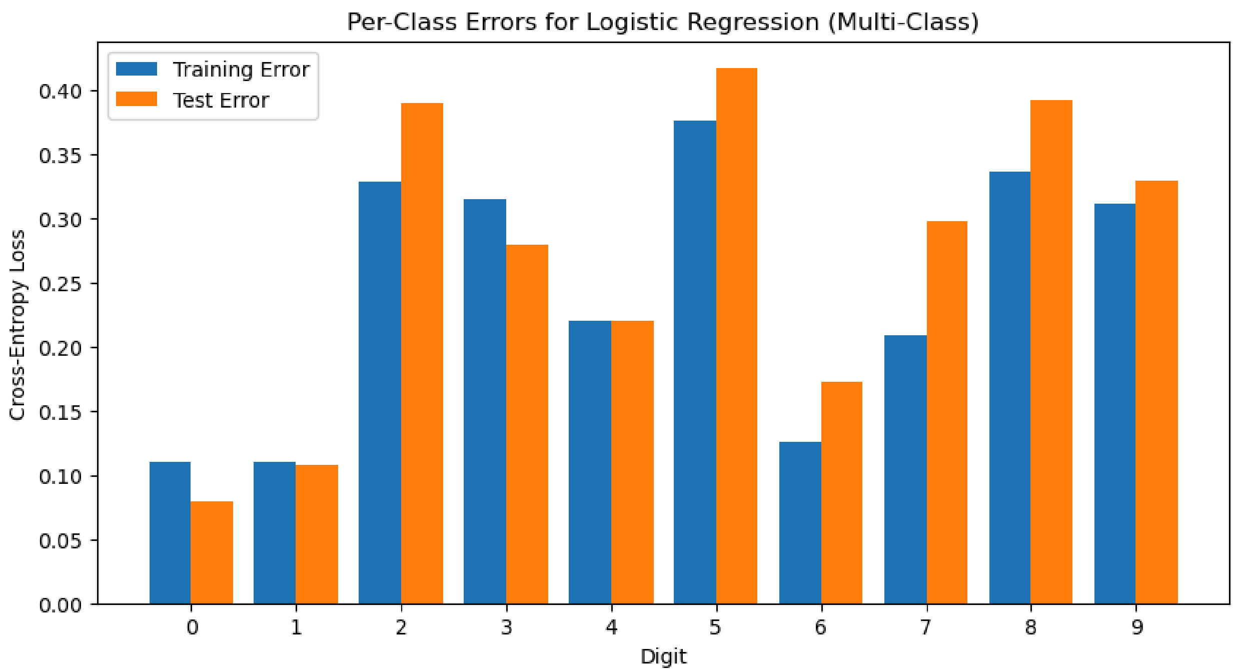


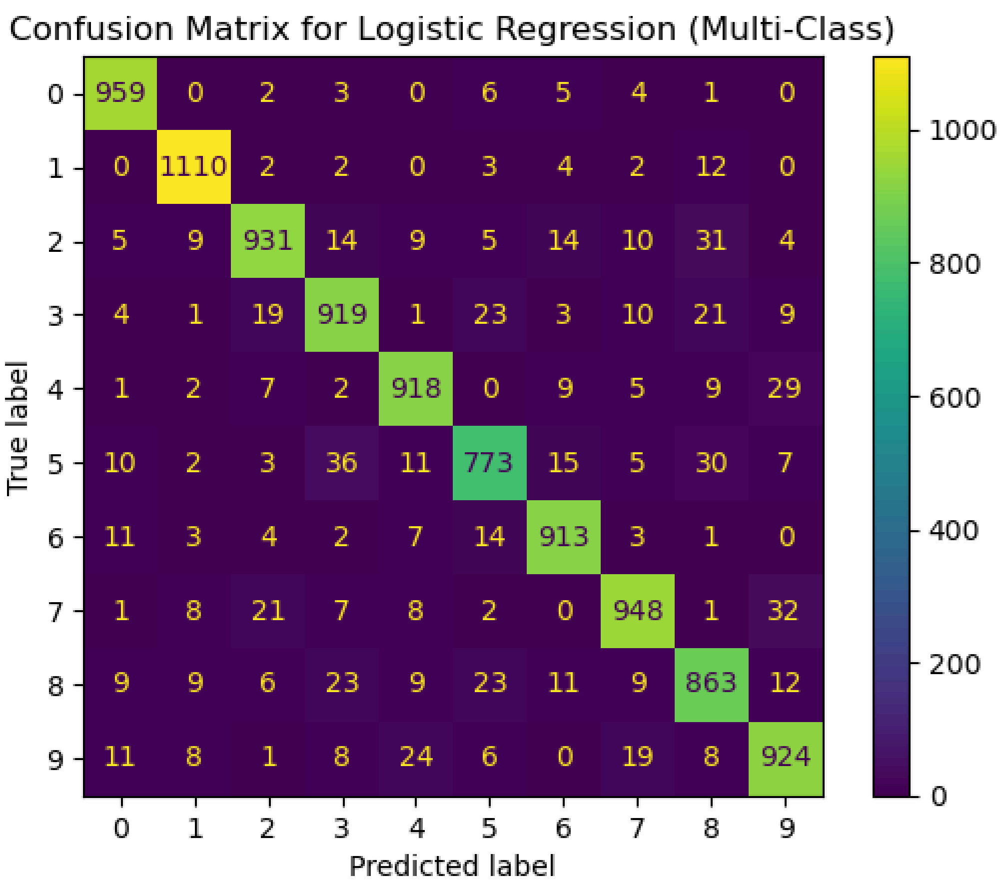
Training Error vs. Test Error

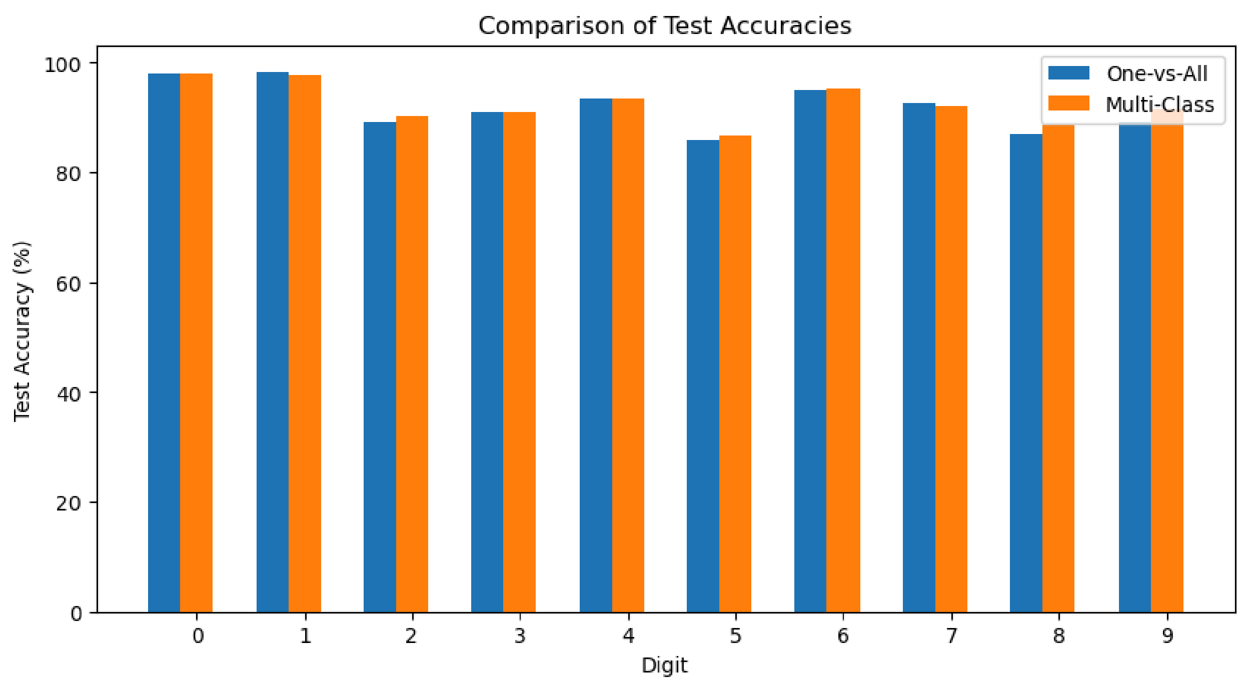
- Observation:
    - Training errors for both strategies (One-vs-All and Multi-class) are consistently lower than test errors.
    - For instance, the training accuracy for Multi-class Logistic Regression is 93.33%, while the test accuracy is slightly lower at 92.58%.
- Reasons for the Difference:
    - Overfitting: The model learns patterns specific to the training dataset, reducing training error. However, these patterns may not generalize well to unseen data, leading to higher test error.
    - Data Variability: Training data may not fully capture the variations in the test data, especially for digits with overlapping features, such as 2 and 8.
    - Noise and Ambiguity: Digits with poor quality (e.g., smudged or incomplete handwriting) increase test error, as they are harder to classify correctly.


Confusion Matrix for Logistic Regression (One-vs-All)
True label
Predicted label
0
1
2
3
4
5
6
7
8
9
960 0 0 2 1 5 5 3 3 1
0 1115 3 1 0 1 4 1 10 0
9 9 920 19 11 4 12 12 32 4
4 1 19 918 2 20 4 14 18 10
1 3 6 2 917 0 9 2 4 38
11 3 1 38 10 766 17 9 29 8
8 4 7 2 4 19 910 1 3 0
2 9 20 6 9 1 1 951 2 27
13 13 7 19 15 29 9 10 847 12
7 8 1 13 35 12 1 21 11 900
1000
800
600
400
200
0


Per-Class Accuracies for Logistic Regression (Multi-Class)
Training Accuracy
Test Accuracy
Accuracy (%)
Digit
0
20
40
60
80
100
0
1
2
3
4
5
6
7
8
9


Per-Class Errors for Logistic Regression (Multi-Class)
Training Error
Test Error
Cross-Entropy Loss
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0 1 2 3 4 5 6 7 8 9
Digit


Confusion Matrix for Logistic Regression (Multi-Class)
True label
Predicted label
0 1 2 3 4 5 6 7 8 9
0 959 0 2 3 0 6 5 4 1 0
1 0 1110 2 2 0 3 4 2 12 0
2 5 9 931 14 9 5 14 10 31 4
3 4 1 19 919 1 23 3 10 21 9
4 1 2 7 2 918 0 9 5 9 29
5 10 2 3 36 11 773 15 5 30 7
6 11 3 4 2 7 14 913 3 1 0
7 1 8 21 7 8 2 0 948 1 32
8 9 9 6 23 9 23 11 9 863 12
9 11 8 1 8 24 6 0 19 8 924
0
200
400
600
800
1000



Comparison of Test Accuracies
One-vs-All
Multi-Class
Test Accuracy (%)
0
20
40
60
80
100
0
1
2
3
4
5
6
7
8
9
Digit

SVM

--------------SVM-------------------

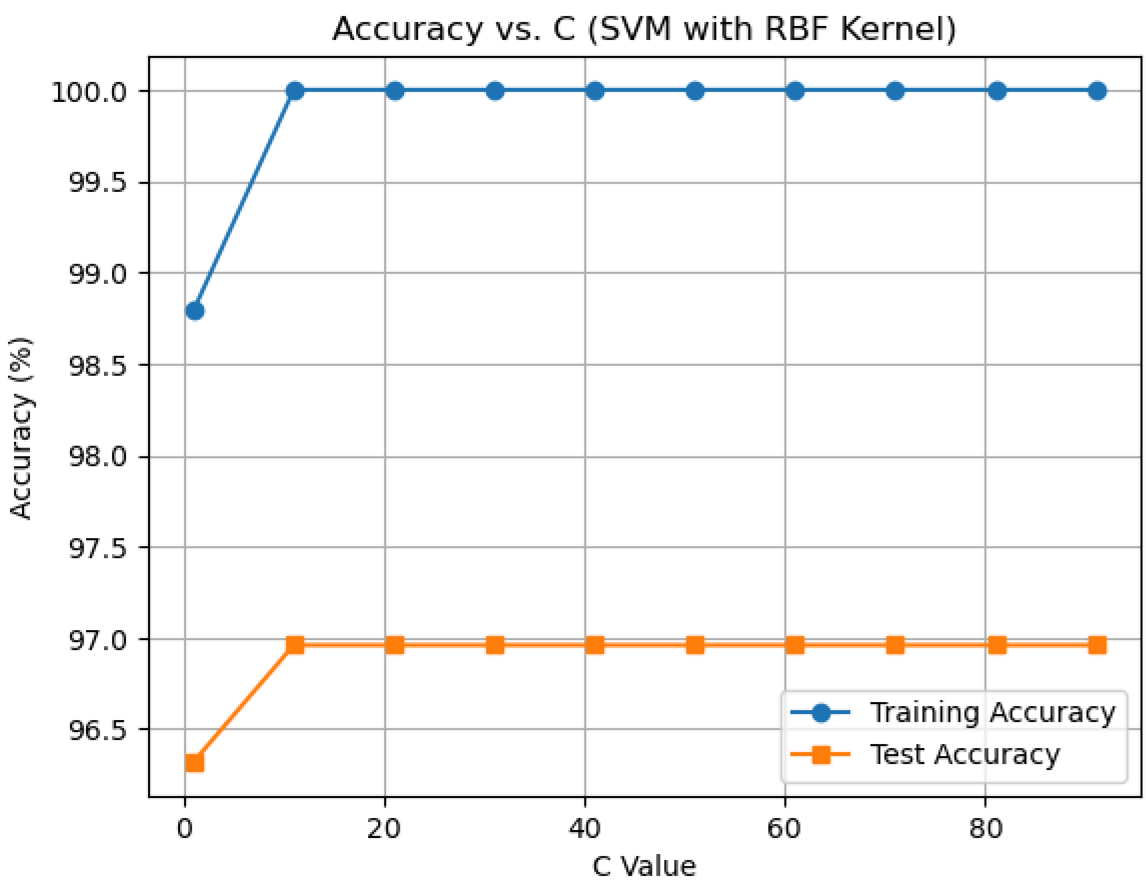
Training SVM with Linear Kernel...
Training Accuracy: 99.75 %
Validation Accuracy: 91.64 %
Testing Accuracy: 91.97999999999999 %

Training SVM with RBF Kernel (gamma=1)...
Training Accuracy: 100.0 %
Validation Accuracy: 18.55 %
Testing Accuracy: 20.06 %

Training SVM with RBF Kernel (default gamma)...
Training Accuracy: 98.79 %
Validation Accuracy: 96.27 %
Testing Accuracy: 96.32 %

Training SVM with RBF Kernel (default gamma) and varying C...
Training with C = 1...
Training Accuracy: 98.79%
Validation Accuracy: 96.27%
Testing Accuracy: 96.32%
Training with C = 10...
Training Accuracy: 100.0%
Validation Accuracy: 96.8%
Testing Accuracy: 96.95%
Training with C = 20...
Training Accuracy: 100.0%
Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%
Training with C = 30...
Training Accuracy: 100.0%
Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%
Training with C = 40...
Training Accuracy: 100.0%

Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%
Training with C = 50...
Training Accuracy: 100.0%
Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%
Training with C = 70...
Training Accuracy: 100.0%
Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%
Training with C = 100...
Training Accuracy: 100.0%
Validation Accuracy: 96.78999999999999%
Testing Accuracy: 96.96000000000001%



Linear Kernel:

- Testing Accuracy: 91.98%
- The linear kernel achieves high accuracy, indicating that the MNIST dataset is relatively well-separated in its original feature space. However, it is not as effective as the RBF kernel for capturing more complex relationships.

RBF Kernel with Gamma=1:

- Testing Accuracy: 20.06%
- The model performs poorly with gamma=1, likely due to overfitting. A high gamma value focuses too much on individual data points, leading to poor generalization on unseen data.

RBF Kernel with Default Gamma:

- Testing Accuracy: 96.32%
- This configuration achieves the best performance, demonstrating the RBF kernel's ability to capture nonlinear patterns while maintaining generalization.

RBF Kernel with Default Gamma and Varying C:

- Validation Accuracies:
  - C=1: 96.27%
  - C=10: 96.80% (Peak accuracy)
  - C=20 to C=100: 96.79%
- Increasing C improves accuracy up to a certain point (C=10). Beyond this, the accuracy stabilizes, indicating diminishing returns.

Linear Kernel:

- The linear kernel's performance (91.98%) reflects its simplicity. It is computationally efficient but lacks the flexibility to model complex nonlinear relationships. This kernel is suitable for datasets that are linearly separable or close to it.

RBF Kernel:

- The RBF kernel with gamma=1 performed poorly (20.06%) due to its high sensitivity to individual data points. This setting caused the model to overfit the training data, reducing its ability to generalize.

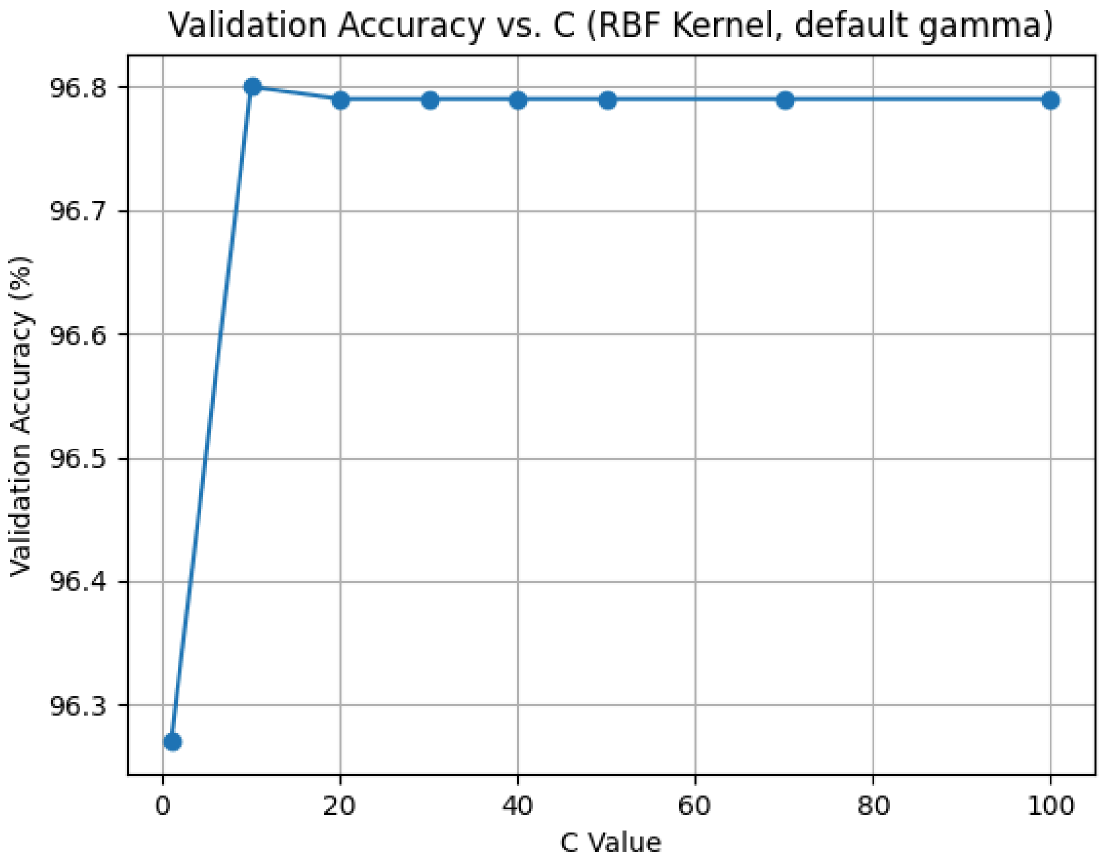
- The RBF kernel with default gamma achieved the best performance (96.32%) by balancing complexity and generalization.

Impact of Regularization Parameter (C):

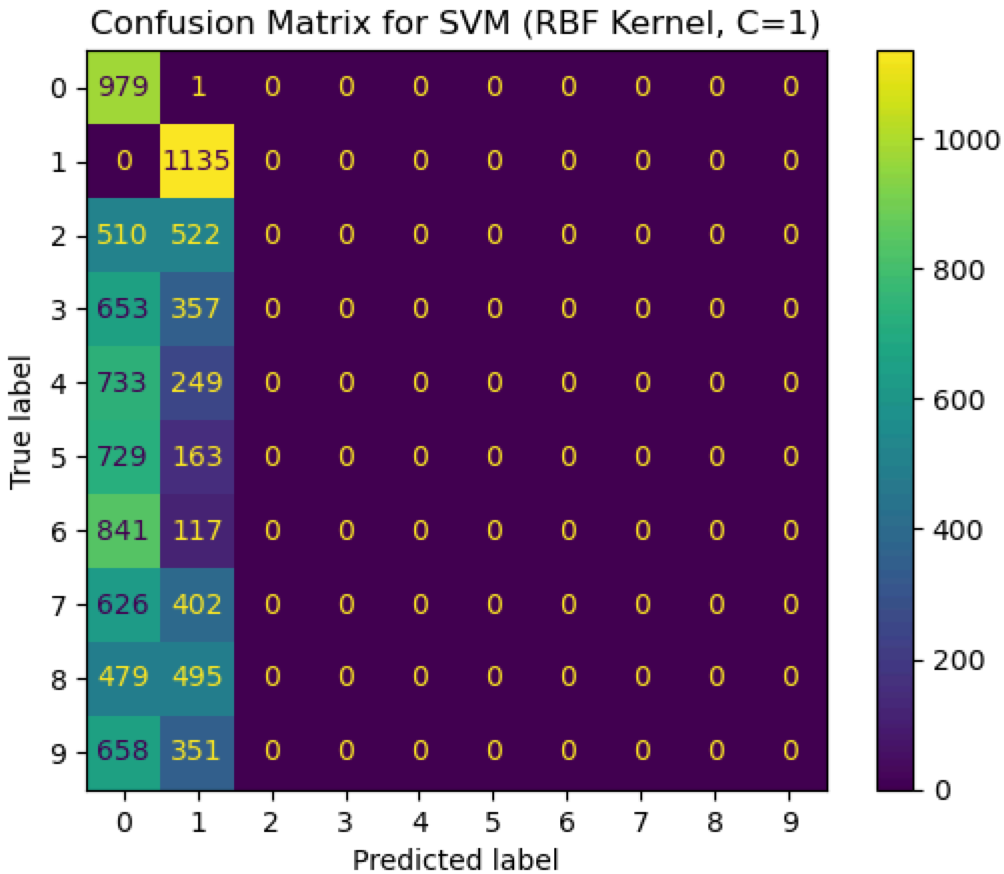
- The C parameter controls the tradeoff between maximizing the margin and minimizing classification error.
- At lower values (C=1), the model underfits slightly, achieving a validation accuracy of 96.27%.
- At higher values (C=10), the accuracy peaks (96.80%), indicating better separation of classes.
- Beyond C=10, the accuracy stabilizes, suggesting that increasing C further does not significantly improve the model's ability to classify data.

Performance Differences:

- Linear Kernel vs. RBF Kernel:
  - The RBF kernel outperforms the linear kernel, especially with default gamma. This highlights the importance of nonlinearity in the MNIST dataset.
- Gamma Settings:
  - A high gamma value (gamma=1) leads to poor generalization due to overfitting.
  - The default gamma setting provides the best results, balancing the model's ability to capture patterns and generalize to unseen data.



Validation Accuracy vs. C (RBF Kernel, default gamma)
Validation Accuracy (%)
96.8
96.7
96.6
96.5
96.4
96.3
0
20
40
60
80
100
C Value



1. The RBF kernel with default gamma and C=10 provides the best configuration for the MNIST dataset, achieving:
    - Validation Accuracy: 96.80%
    - Testing Accuracy: 96.32%
2. The linear kernel offers a simpler alternative with reasonable performance (91.98%) but lacks the flexibility of the RBF kernel.
3. Tuning the C parameter is crucial. For this dataset, increasing C beyond 10 offers minimal improvement in accuracy.
4. The RBF kernel is highly sensitive to gamma. Using default gamma achieves a good balance, while a fixed high value (gamma=1) leads to overfitting.