ASSIGNMENT 4

CS5691 Pattern Recognition and Machine Learning

CS5691 Assignment 4

Team Members:

CS24M039	Sagar Biswas
CS24M043	Saurabh Kumar Sahu

Indian Institute of Technology, Madras

Exercise-1: Classifier for Dataset 1

Using linear kernel based SVM :-

Dataset 1: 2-d data: Linearly separable data (Same as Dataset 1 of Assignment 2)

Which means that the data points can be perfectly separated by a straight line (hyperplane in higher dimensions).

Table: 1.1: A table of classification accuracies for the training, validation and test data

Dataset	**Accuracy**
Training	0.986905
Validation	0.987500
Test	0.991667

Table: 1.2 Confusion matrices for training data and test data

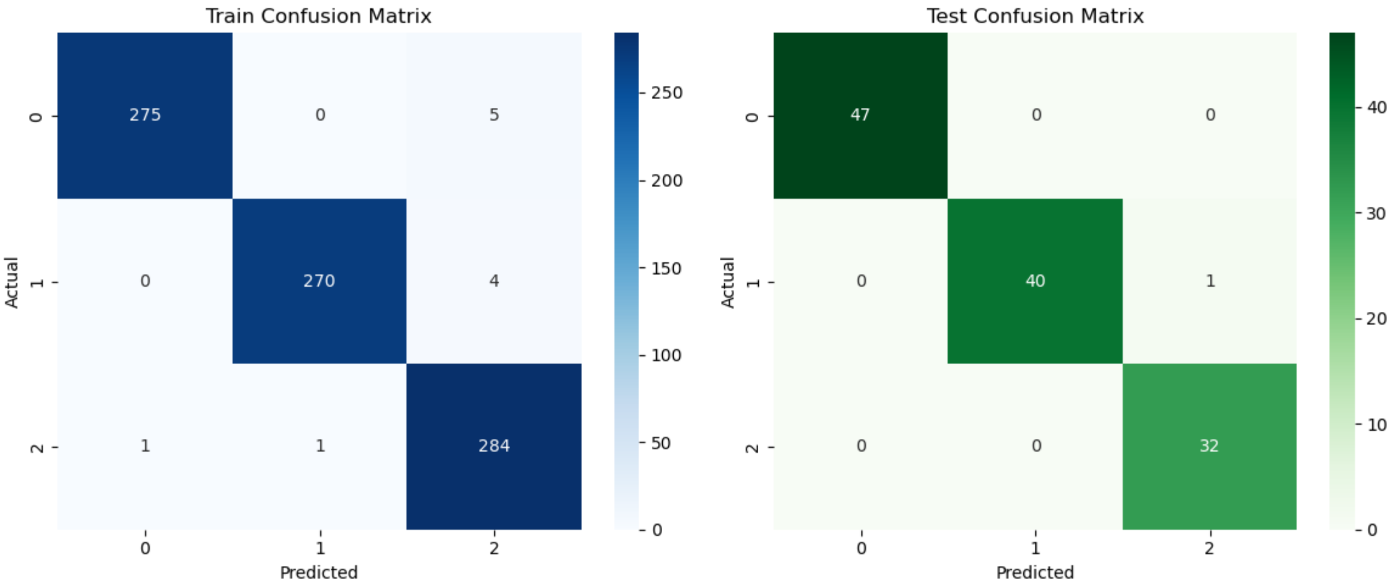


Fig-1.1

Table-1.3: Plots the decision region for the best performing model

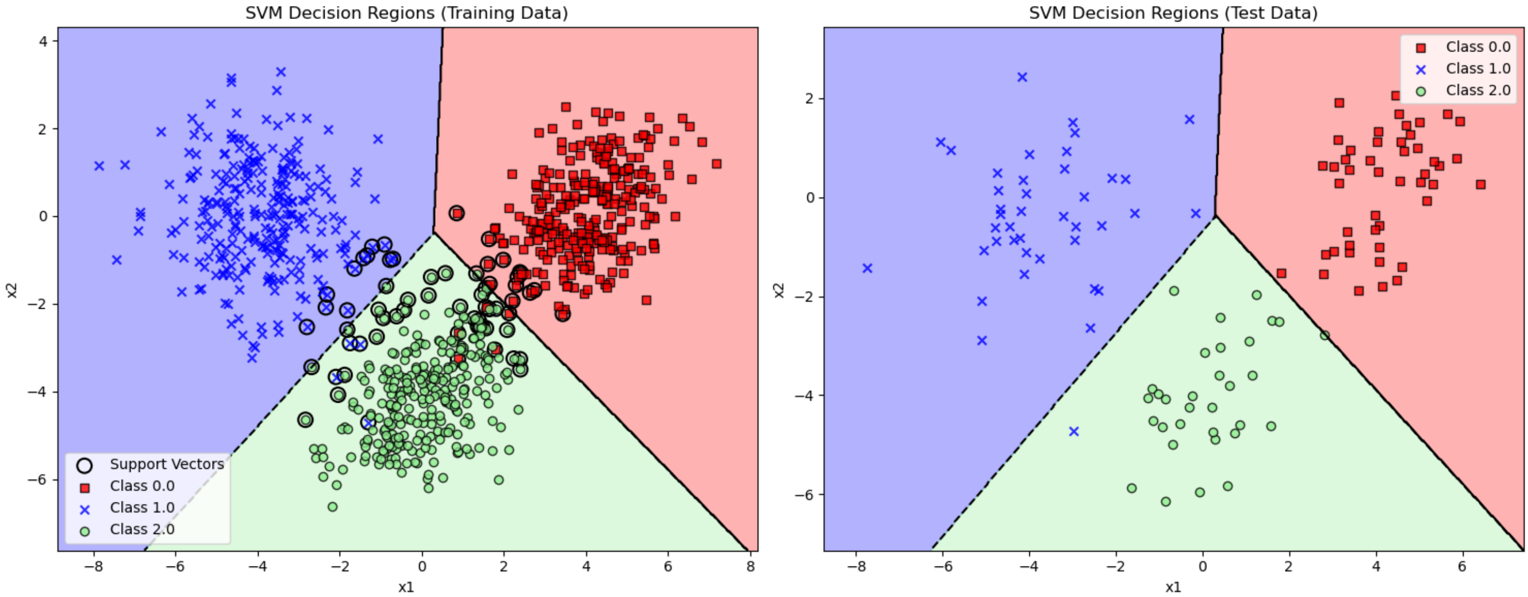


Fig-1.2

Observations:

High Accuracy Across All Sets:

- The SVM classifier shows consistently high accuracy across the training, validation, and test datasets, with values exceeding 98%. This indicates that the model is effectively learning the underlying patterns in the data.

Minimal Overfitting:

- The training accuracy (0.986905) and validation accuracy (0.987500) are very close, suggesting minimal overfitting. The slight drop in training accuracy compared to validation indicates that the model generalizes well to unseen data.

Performance on Test Set:

- The test accuracy (0.991667) is slightly higher than both training and validation accuracies. This could indicate that the model has performed particularly well on the test dataset, potentially due to its characteristics being slightly more aligned with the training data.

Conclusion:

The linear kernel-based SVM has proven to be an effective classifier for Dataset 1, achieving high accuracy and demonstrating robust performance across different datasets. The decision region plots and confusion matrices support these findings, indicating a well-performing model that generalizes well to unseen data. Further analysis could involve testing other kernels or hyperparameter tuning to potentially enhance performance even further.

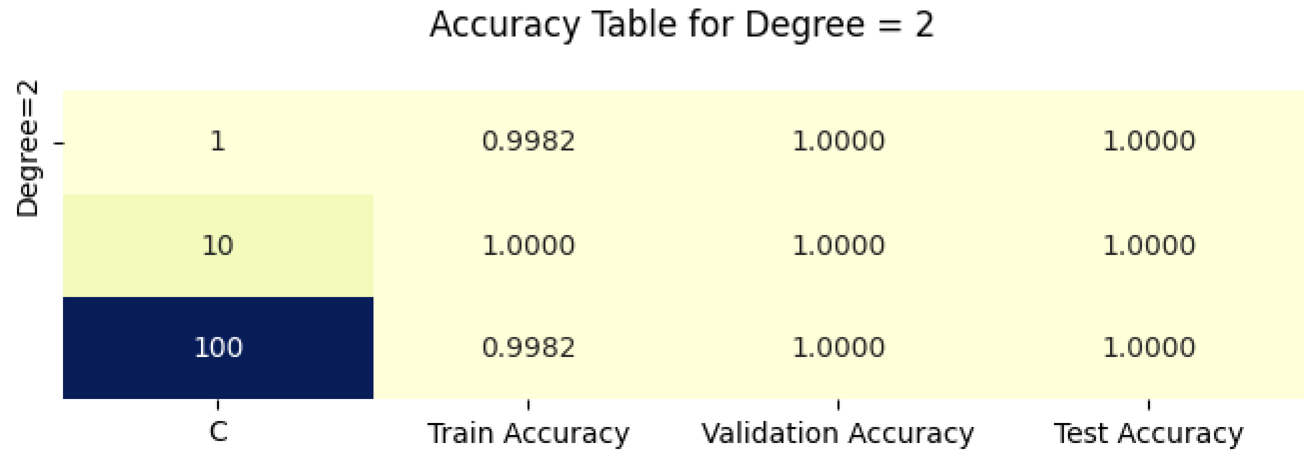
Exercise-2: Classifier for Dataset 2

a) Using polynomial kernel based SVM-:

Dataset 2: 2-d data: Nonlinearly separable data for 2 classes (Same as Dataset 2 of Assignment 2)

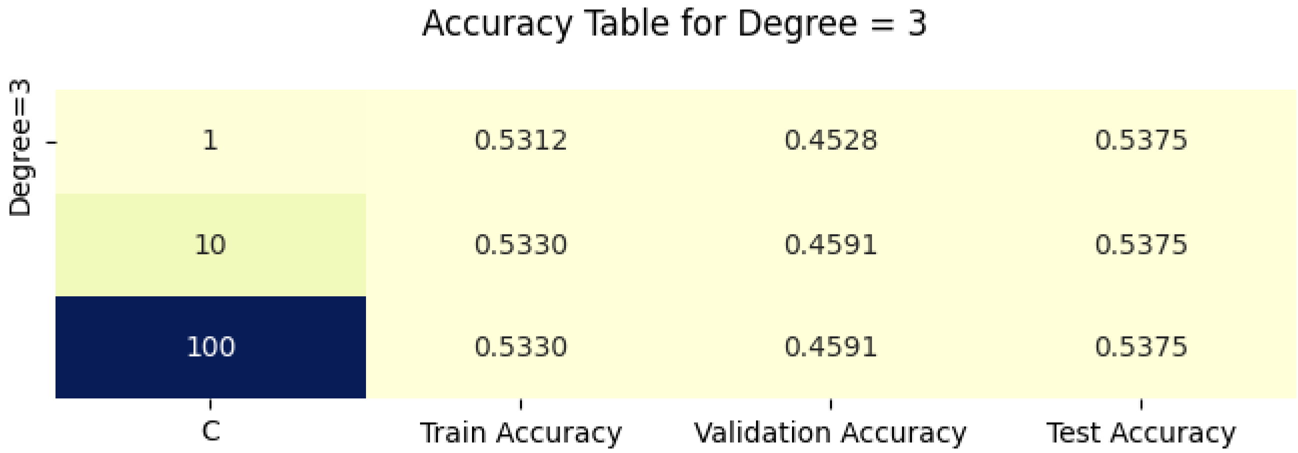
We performed the model training for 4 different values of kernel hyperparameter (degree of polynomial) for d= 2,3,4,5 and C values 1,10 and 100.

Table: 2.1: A table of classification accuracies for the training, validation and test data



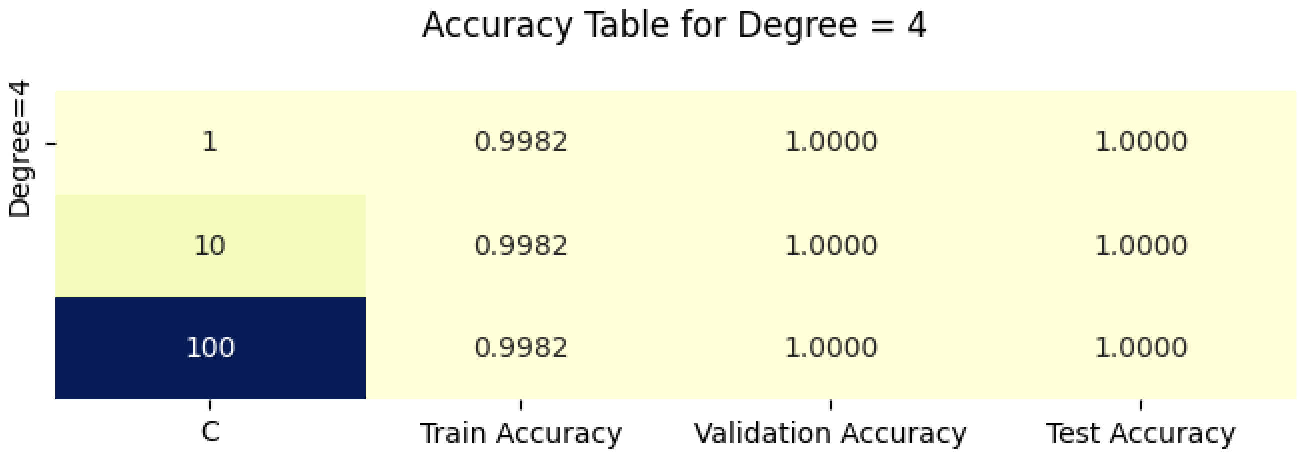
Accuracy Table for Degree = 2

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=2	1	0.9982	1.0000	1.0000
	10	1.0000	1.0000	1.0000
	100	0.9982	1.0000	1.0000



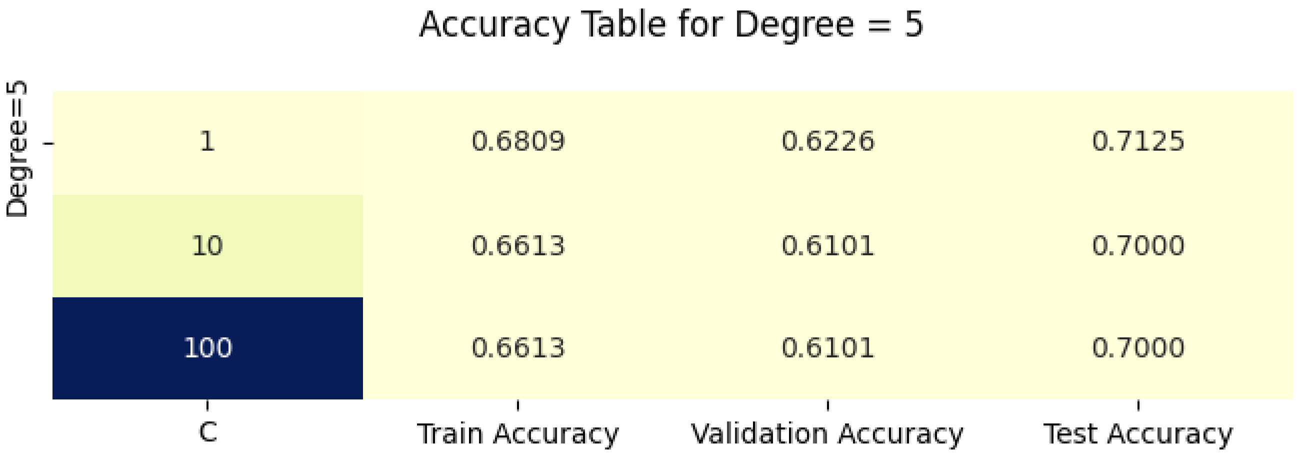
Accuracy Table for Degree = 3

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=3	1	0.5312	0.4528	0.5375
	10	0.5330	0.4591	0.5375
	100	0.5330	0.4591	0.5375



Accuracy Table for Degree = 4

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=4	1	0.9982	1.0000	1.0000
	10	0.9982	1.0000	1.0000
	100	0.9982	1.0000	1.0000



Accuracy Table for Degree = 5

Degree=5	C	Train Accuracy	Validation Accuracy	Test Accuracy
	1	0.6809	0.6226	0.7125
	10	0.6613	0.6101	0.7000
	100	0.6613	0.6101	0.7000

We choose the best performing model based on the model performance on the validation date over all degree of polynomial and C values. from above Table we can see that

The best degree is: 2

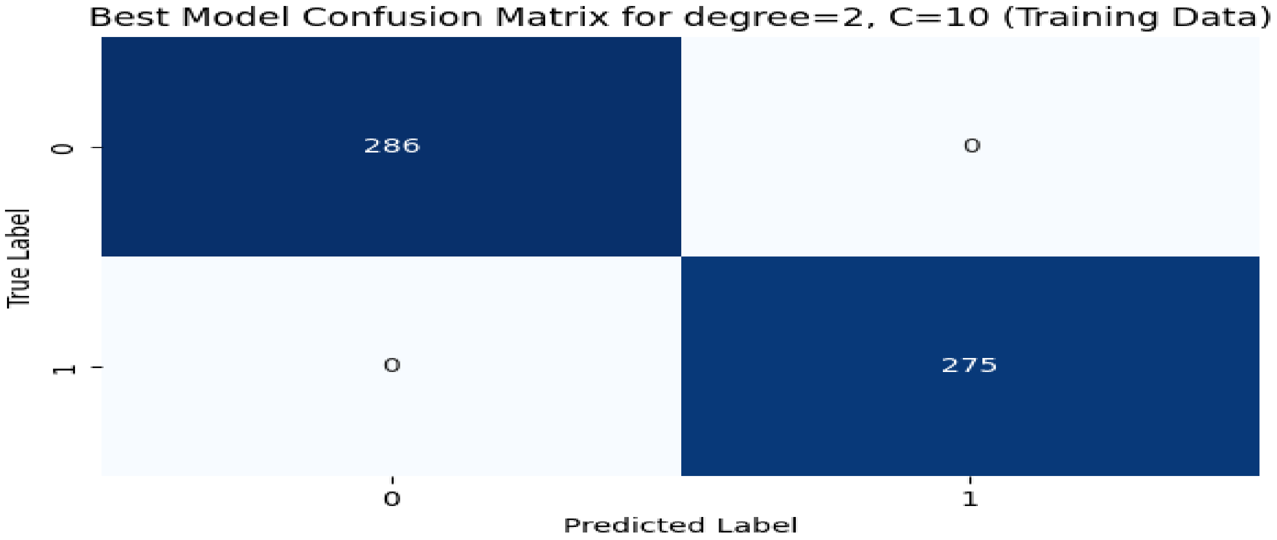
The best C value is: 10

Validation Accuracy: 1.0

Test Accuracy: 1.0

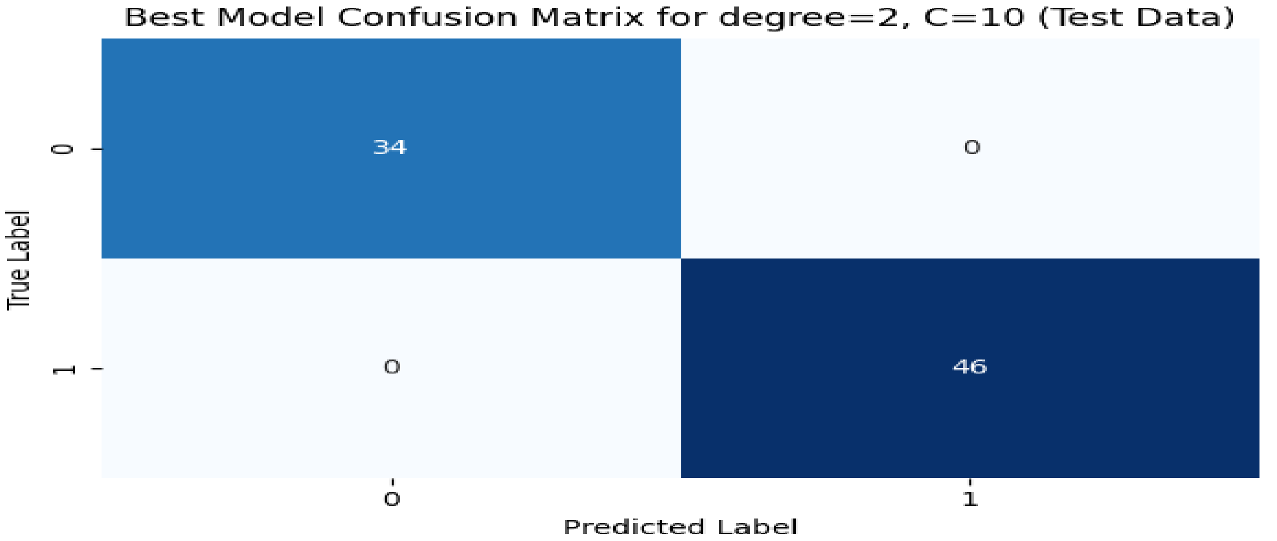
Train Accuracy: 1.0

Table: 2.2 Confusion matrices for training data and test data of the best performing model



Best Model Confusion Matrix for degree=2, C=10 (Training Data)

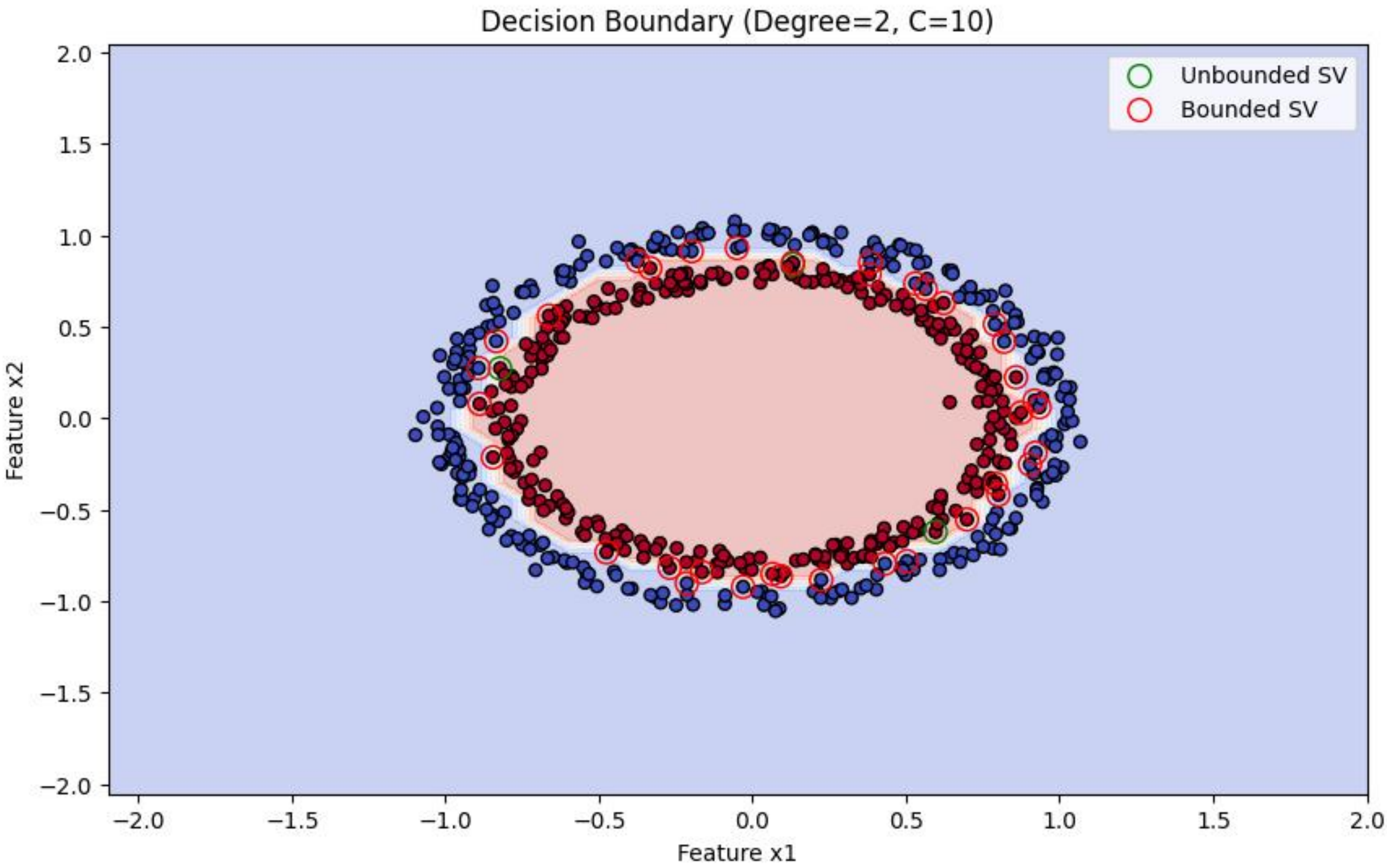
True Label \ Predicted Label	0	1
0	286	0
1	0	275



Best Model Confusion Matrix for degree=2, C=10 (Test Data)

True Label \ Predicted Label	0	1
0	34	0
1	0	46

Table-2.3: Plots the decision region for the best performing model



Observations:

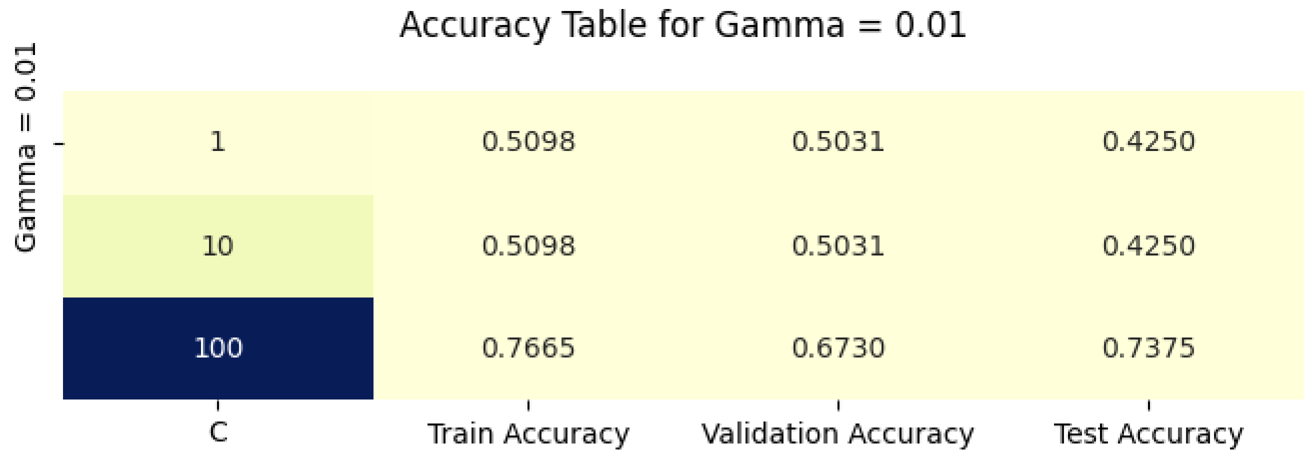
- The decision boundary forms an elliptical region around the data points, which is expected for a polynomial kernel of degree 2.
- This indicates that the model has captured the non-linear separation required for this dataset, where a simple linear boundary would not suffice.
- Unbounded Support Vectors (green circles): These lie exactly on the margin. There are fewer unbounded support vectors compared to bounded ones, as they only lie along the margin without violating it.
- Bounded Support Vectors (red circles): These are within the margin or slightly misclassified, and are closer to the boundary or even inside the opposite class region.
- The presence of many bounded support vectors indicates that the data points are quite close to the boundary, especially in the denser regions of the distribution.

b) Using Gaussian kernel based SVM-:

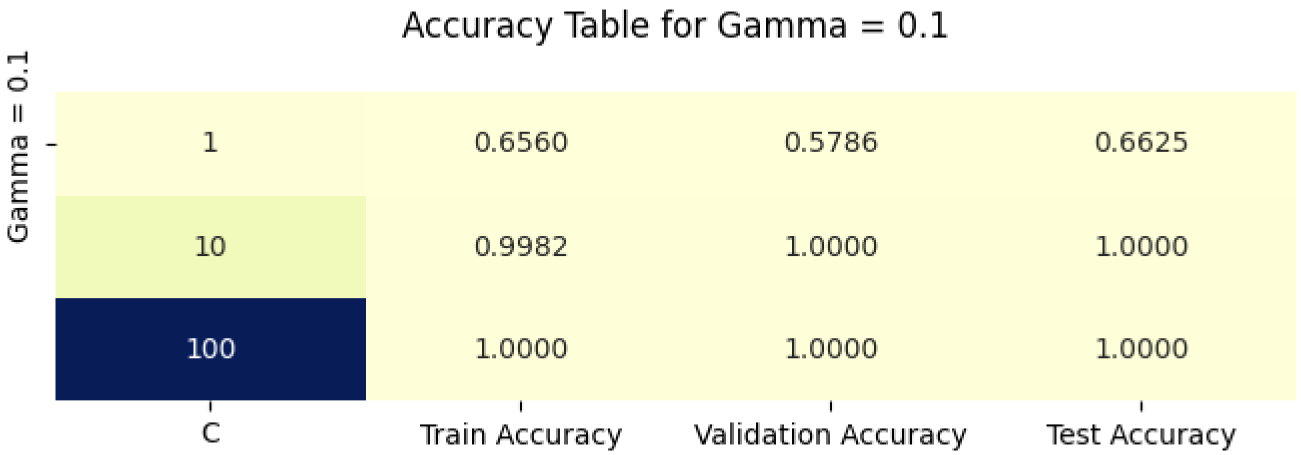
Dataset 2: 2-d data: Nonlinearly separable data for 2 classes (Same as Dataset 2 of Assignment 2)

We performed the model training for 4 different values of kernel hyperparameter (Kernel width) for gamma values of 0.01, 0.1, 1, 10 and C values 1,10 and 100.

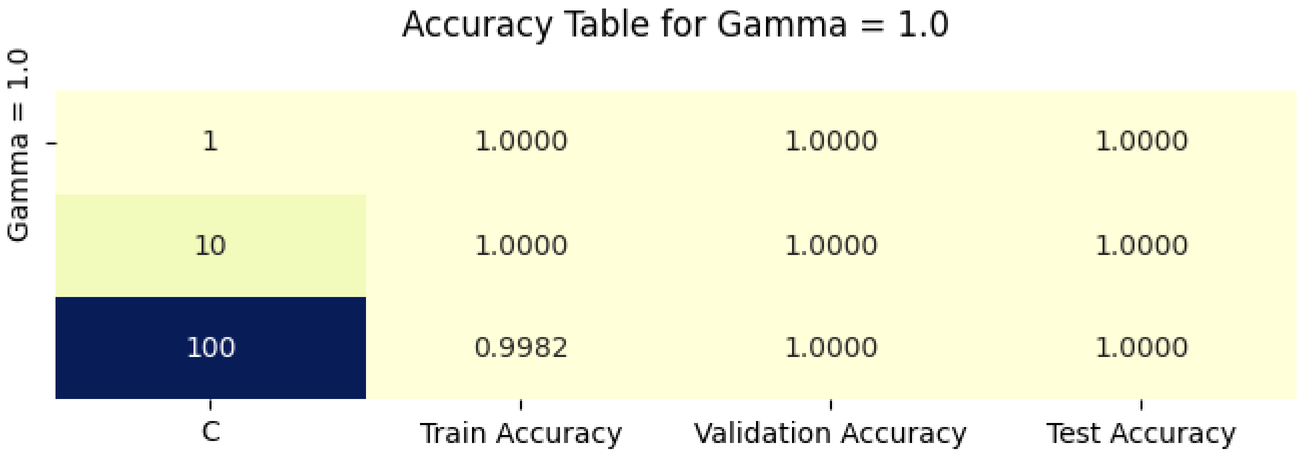
Table: 2.3: A table of classification accuracies for the training, validation and test data


Accuracy Table for Gamma = 0.01

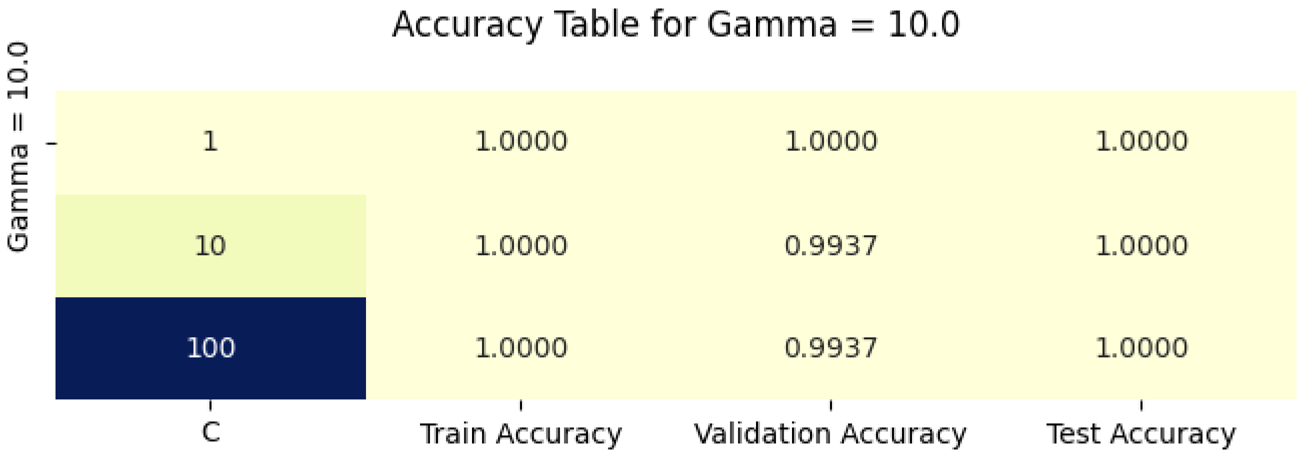
C	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.5098	0.5031	0.4250
10	0.5098	0.5031	0.4250
100	0.7665	0.6730	0.7375


Accuracy Table for Gamma = 0.1

C	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.6560	0.5786	0.6625
10	0.9982	1.0000	1.0000
100	1.0000	1.0000	1.0000


Accuracy Table for Gamma = 1.0

C	Train Accuracy	Validation Accuracy	Test Accuracy
1	1.0000	1.0000	1.0000
10	1.0000	1.0000	1.0000
100	0.9982	1.0000	1.0000



Accuracy Table for Gamma = 10.0

Gamma = 10.0	C	Train Accuracy	Validation Accuracy	Test Accuracy
	1	1.0000	1.0000	1.0000
	10	1.0000	0.9937	1.0000
	100	1.0000	0.9937	1.0000

We choose the best performing model based on the model performance on the validation date over all degree of polynomial and C values. from above Table we can see that

The best gamma is: 0.1

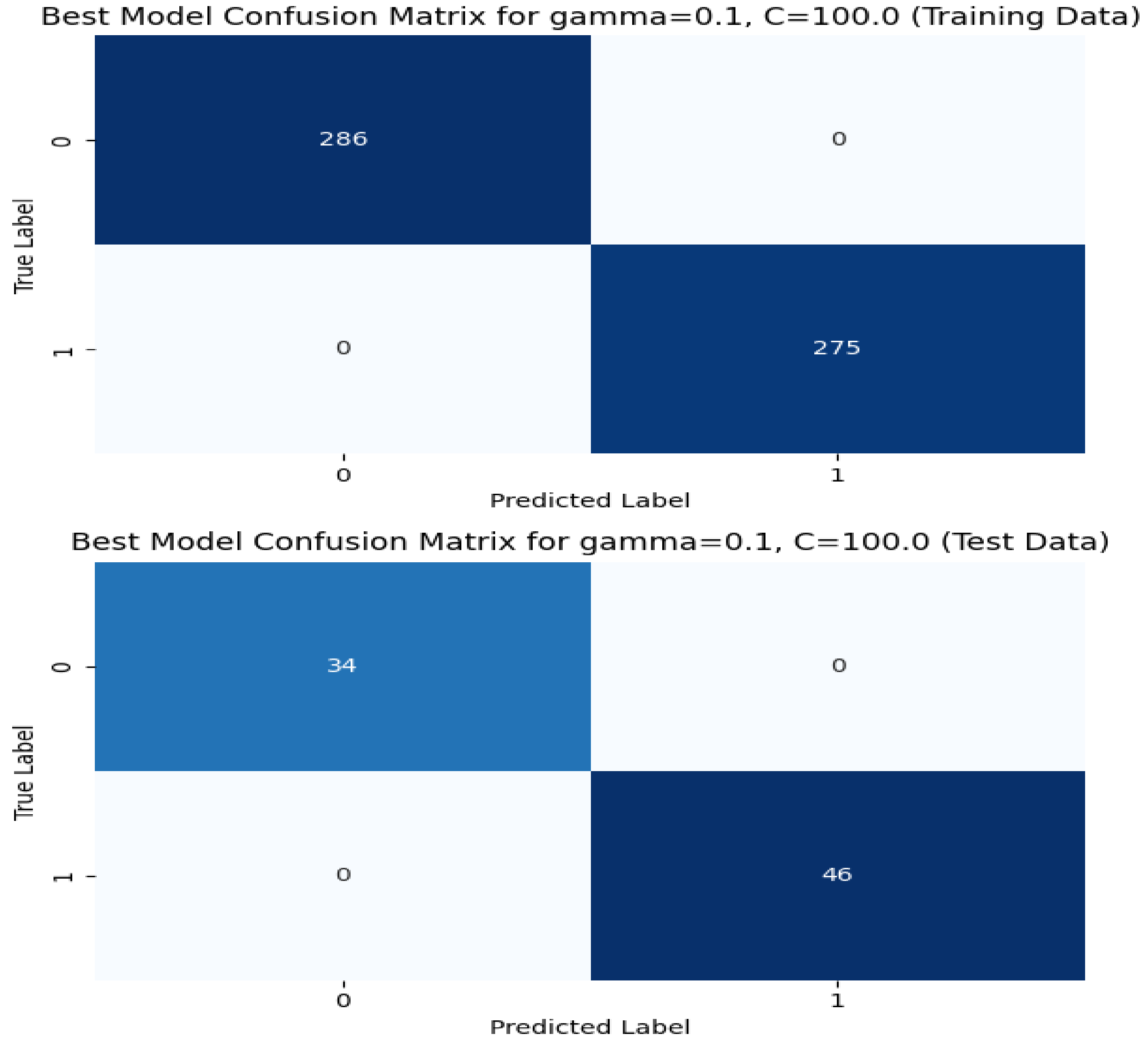
The best C value is: 100.0

Validation Accuracy: 1.0

Test Accuracy: 1.0

Train Accuracy: 1.0

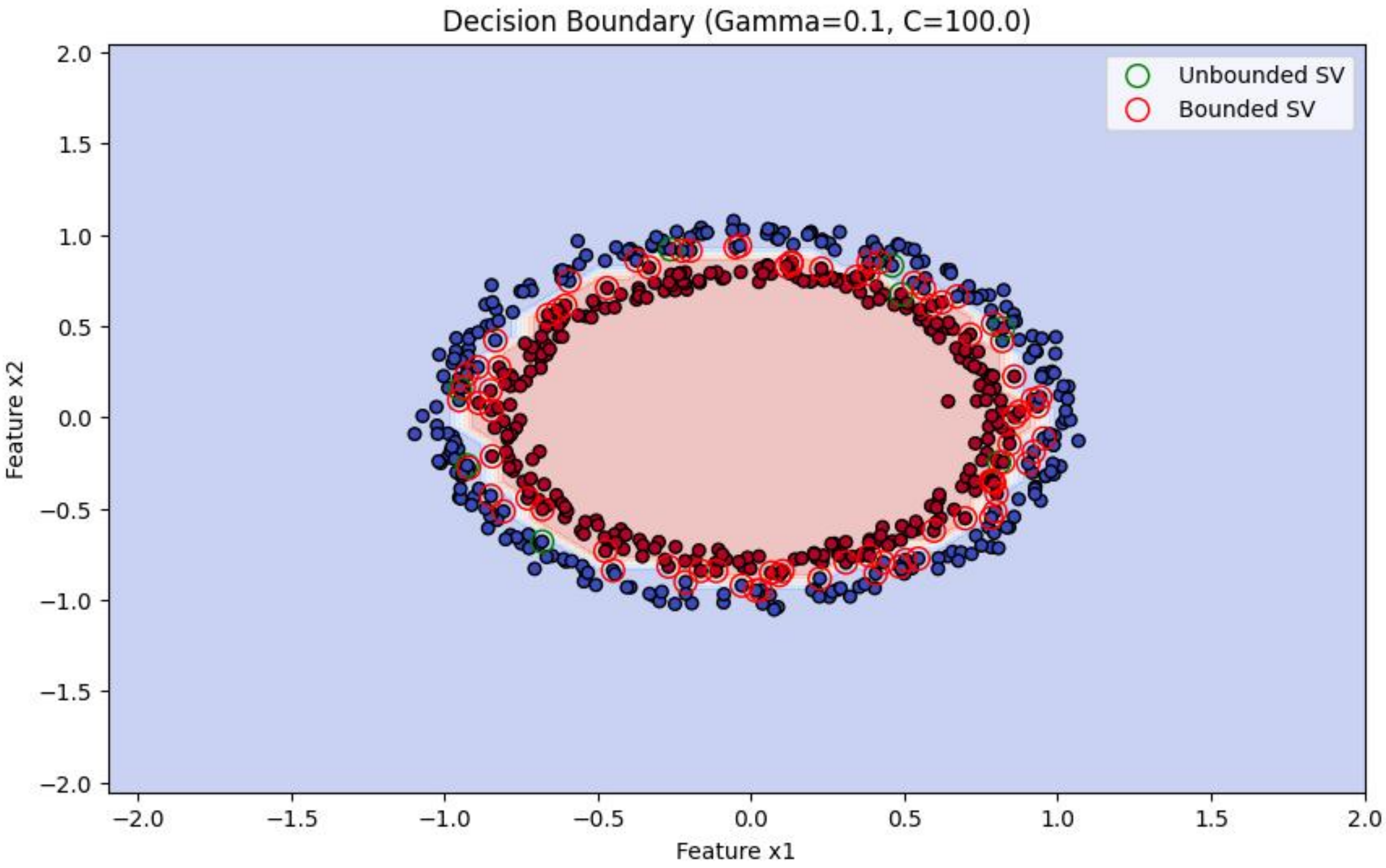
Table: 2.4 Confusion matrices for training data and test data of the best performing model



Best Model Confusion Matrix for gamma=0.1, C=100.0 (Training Data)

Best Model Confusion Matrix for gamma=0.1, C=100.0 (Test Data)

Table-2.5: Plots the decision region for the best performing model



Observations:

- The boundary is smooth and forms a tight oval around the data, capturing the non-linear nature of the classes. The Gaussian kernel allows the SVM to adapt to more complex shapes in the data.
- Unbounded (green): Few in number, lying exactly on the margin.
- Bounded (red): Many near or slightly inside the decision boundary, reflecting the model's high tolerance for misclassified points within this small margin area due to the large CCC value.

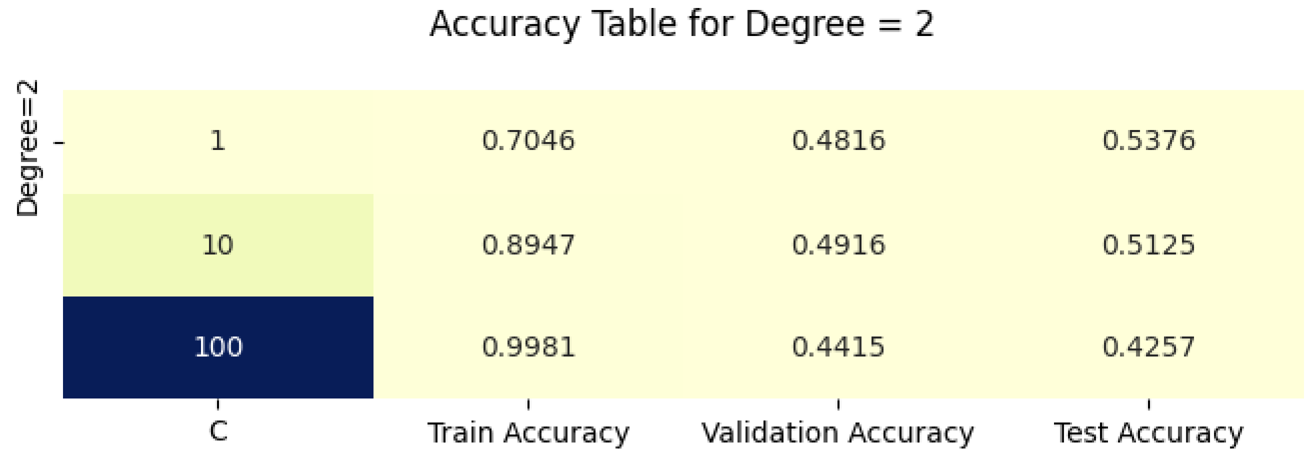
Exercise-3: Classifier for Dataset 3

a) Using polynomial kernel based SVM-:

Dataset 3:: Image data (Dimension of feature vector: 35) for 5 classes (Same as Dataset 3 of Assignment 2)

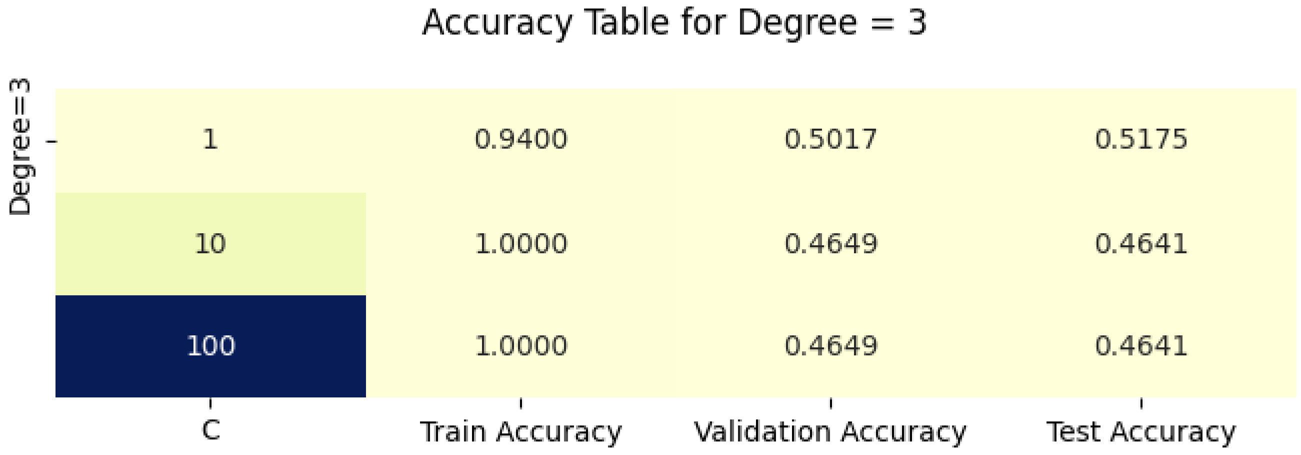
We performed the model training for 4 different values of kernel hyperparameter (degree of polynomial) for d= 2,3,4,5 and C values 1,10 and 100.

Table: 3.1: A table of classification accuracies for the training, validation and test data



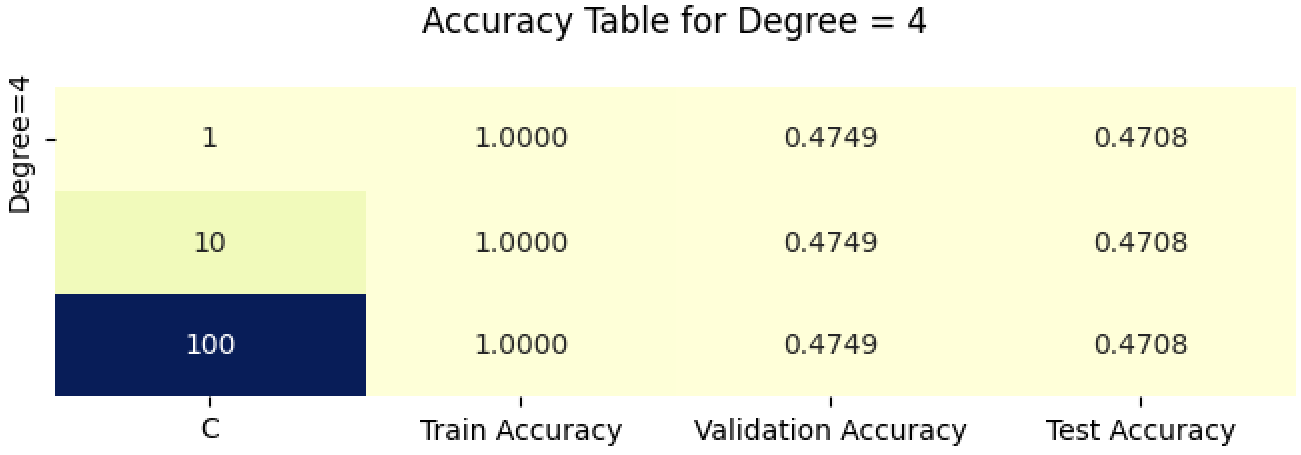
Accuracy Table for Degree = 2

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=2	1	0.7046	0.4816	0.5376
	10	0.8947	0.4916	0.5125
	100	0.9981	0.4415	0.4257



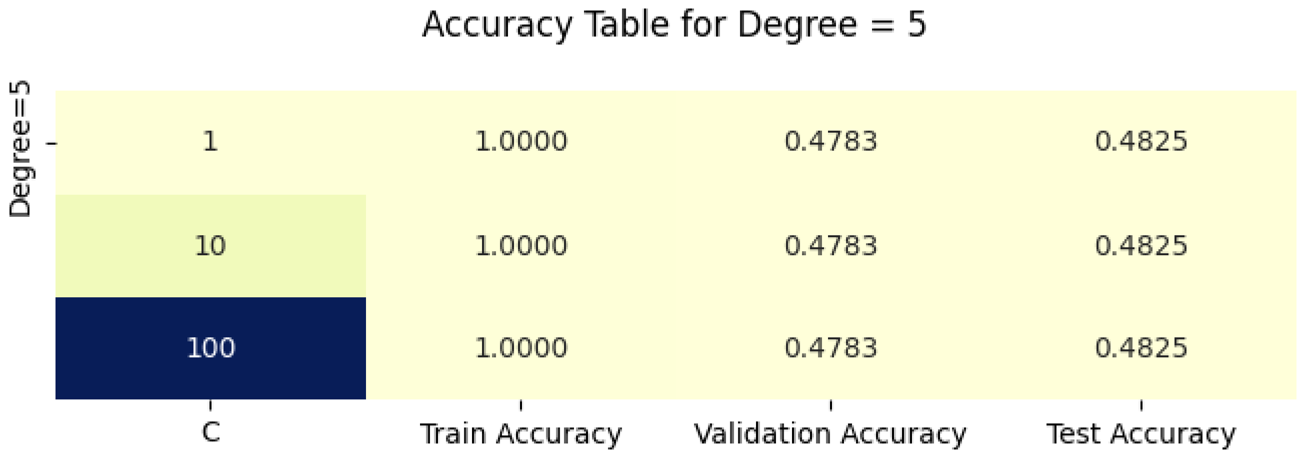
Accuracy Table for Degree = 3

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=3	1	0.9400	0.5017	0.5175
	10	1.0000	0.4649	0.4641
	100	1.0000	0.4649	0.4641



Accuracy Table for Degree = 4

	C	Train Accuracy	Validation Accuracy	Test Accuracy
Degree=4	1	1.0000	0.4749	0.4708
	10	1.0000	0.4749	0.4708
	100	1.0000	0.4749	0.4708


Accuracy Table for Degree = 5

Degree=5 C	Train Accuracy	Validation Accuracy	Test Accuracy
1	1.0000	0.4783	0.4825
10	1.0000	0.4783	0.4825
100	1.0000	0.4783	0.4825

We choose the best performing model based on the model performance on the validation date over all degree of polynomial and C values. from above Table we can see that

The best degree is: 3

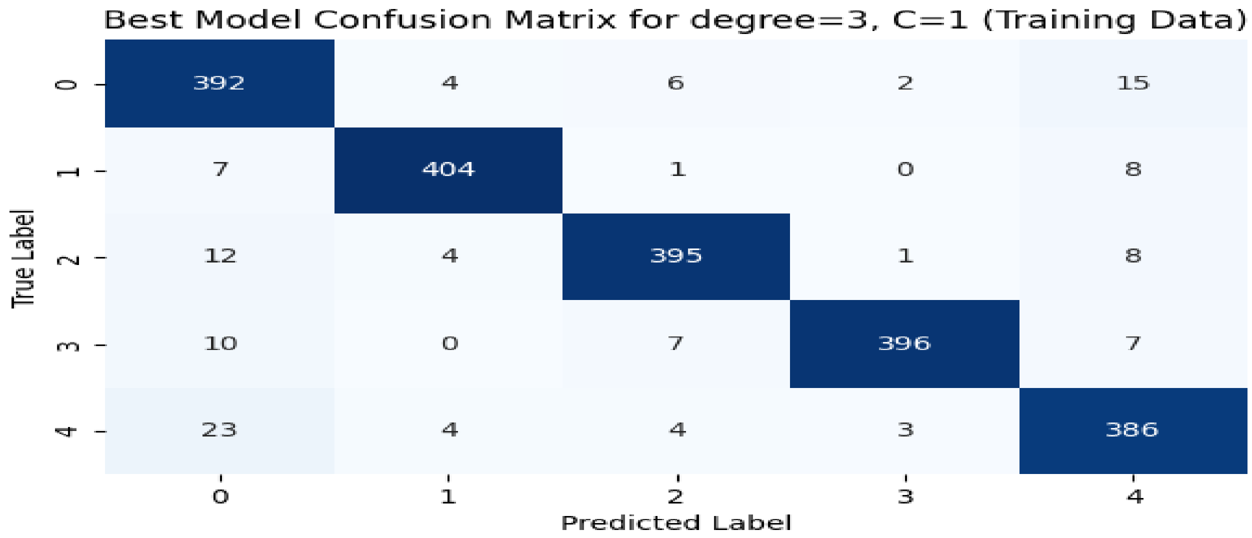
The best C value is: 1

Validation Accuracy: 0.50

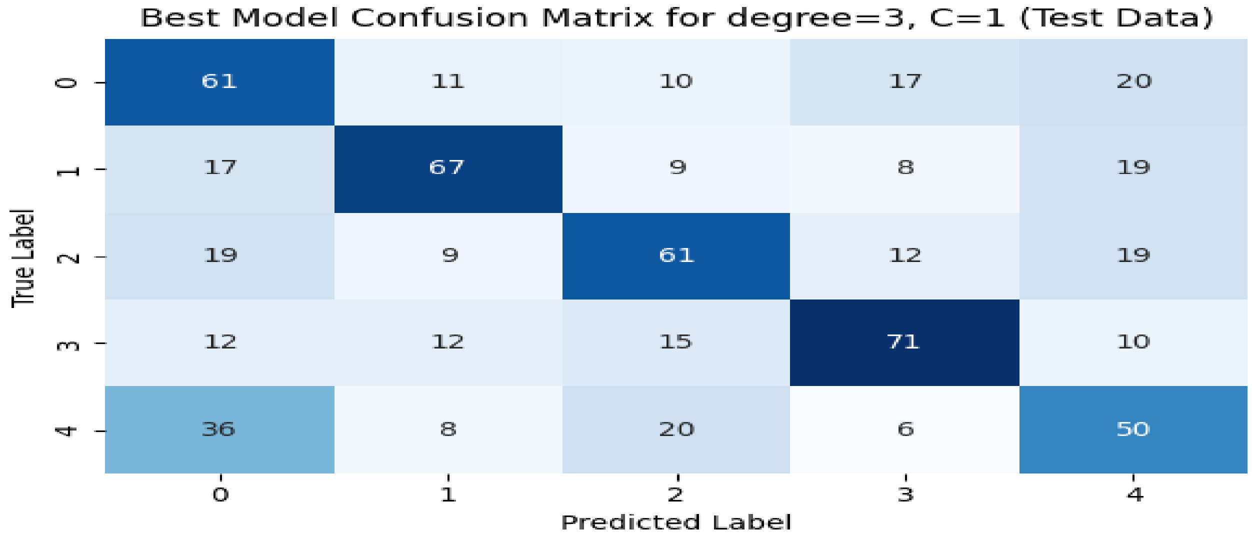
Test Accuracy: 0.51

Train Accuracy: 0.94

Table: 3.2 Confusion matrices for training data and test data of the best performing model


Best Model Confusion Matrix for degree=3, C=1 (Training Data)

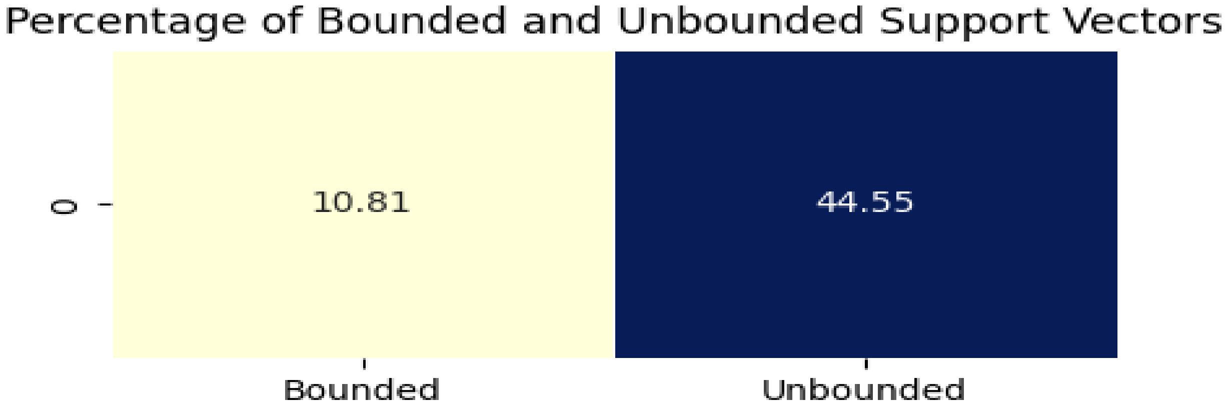
True Label \ Predicted Label	0	1	2	3	4
0	392	4	6	2	15
1	7	404	1	0	8
2	12	4	395	1	8
3	10	0	7	396	7
4	23	4	4	3	386


Best Model Confusion Matrix for degree=3, C=1 (Test Data)

True Label \ Predicted Label	0	1	2	3	4
0	61	11	10	17	20
1	17	67	9	8	19
2	19	9	61	12	19
3	12	12	15	71	10
4	36	8	20	6	50

Table-3.3: A table of percentage of bounded and unbounded support vectors out of the

training examples, for the best performing model



Observations:

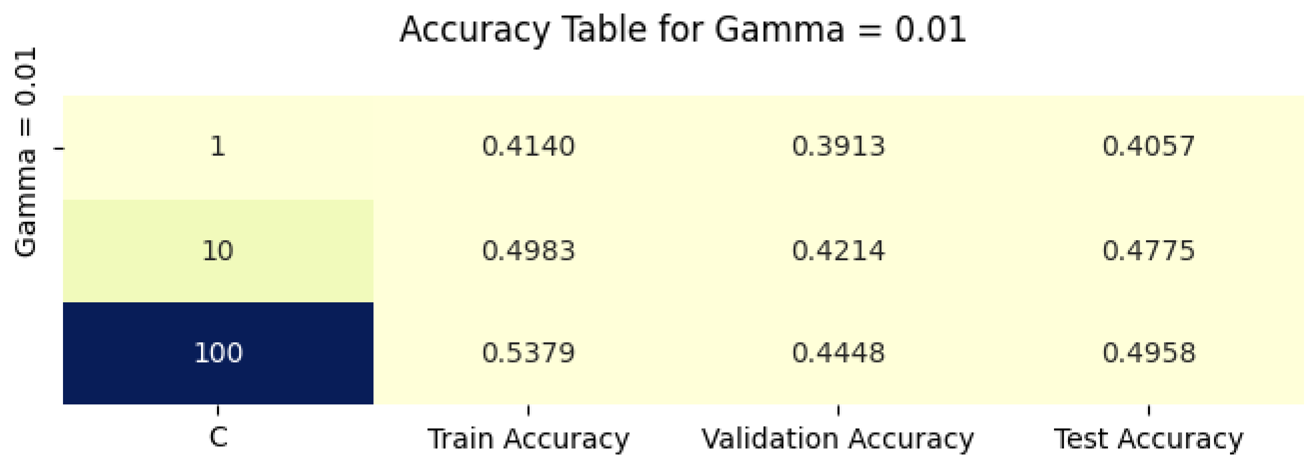
- Bounded Support Vectors (10.81%): These represent samples that the model couldn't push further from the decision boundary due to the constraints imposed by C.
- Unbounded Support Vectors (44.55%): A high proportion of unbounded SVs suggests that many training samples lie close to the margin, which typically indicates a complex decision boundary.

b) Using Gaussian kernel based SVM-:

Dataset 3: Image data (Dimension of feature vector: 35) for 5 classes (Same as Dataset 3 of Assignment 2)

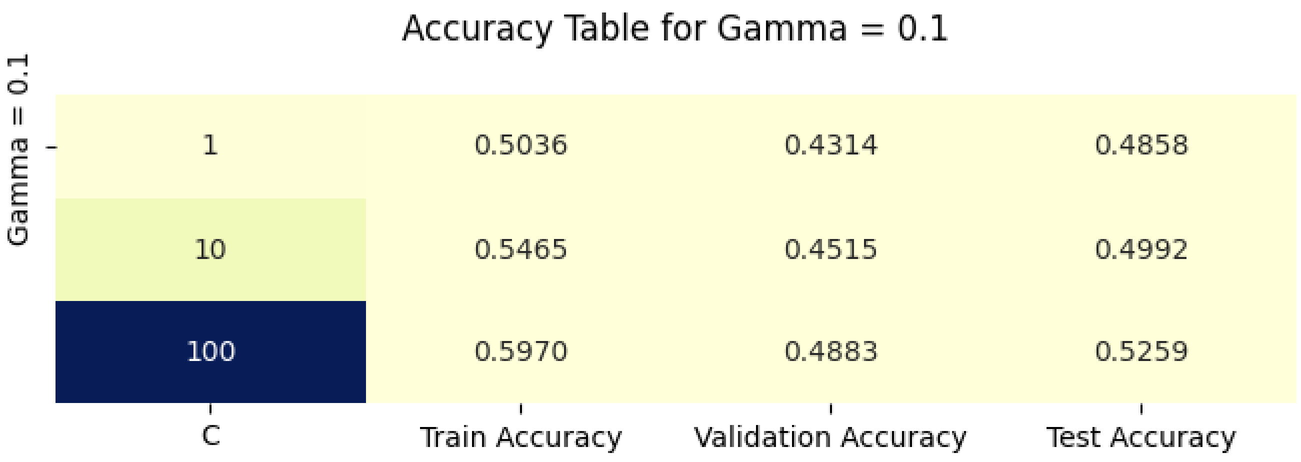
We performed the model training for 4 different values of kernel hyperparameter (Kernel width) for gamma values of 0.01, 0.1, 1, 10 and C values 1,10 and 100.

Table: 3.4: A table of classification accuracies for the training, validation and test data



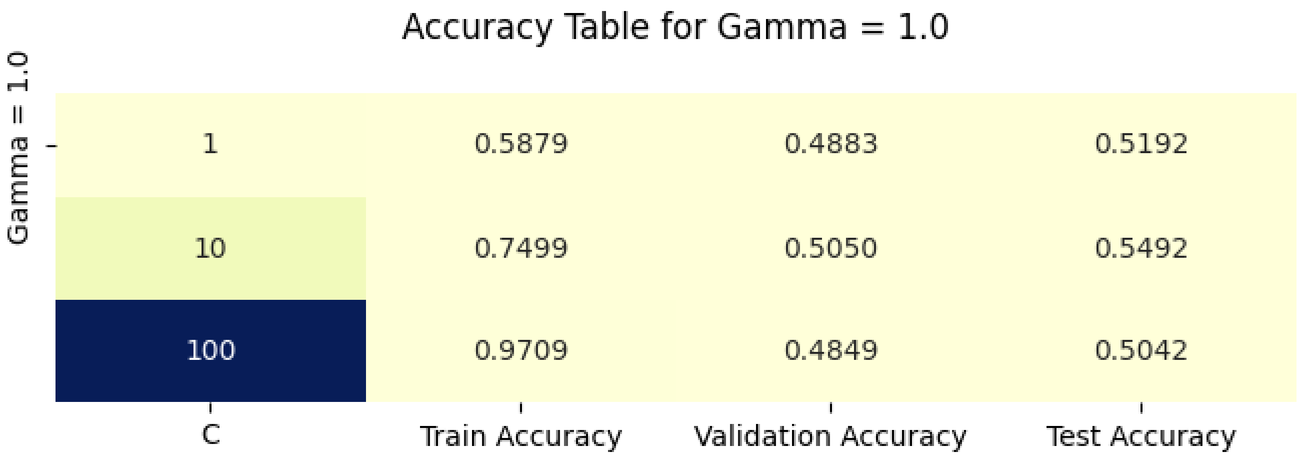
Accuracy Table for Gamma = 0.01

C	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.4140	0.3913	0.4057
10	0.4983	0.4214	0.4775
100	0.5379	0.4448	0.4958



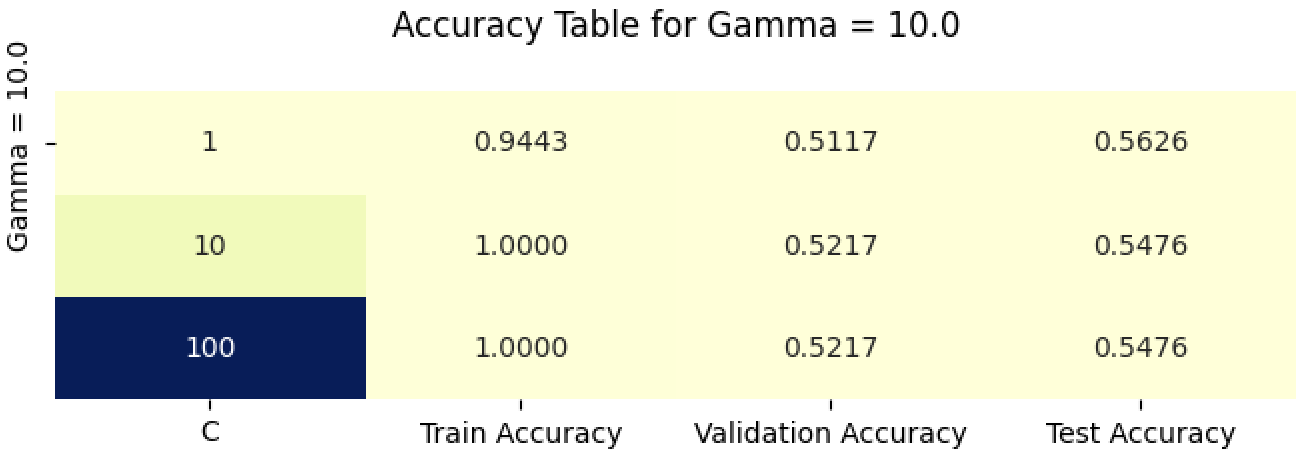
Accuracy Table for Gamma = 0.1

C	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.5036	0.4314	0.4858
10	0.5465	0.4515	0.4992
100	0.5970	0.4883	0.5259



Accuracy Table for Gamma = 1.0

C	Train Accuracy	Validation Accuracy	Test Accuracy
1	0.5879	0.4883	0.5192
10	0.7499	0.5050	0.5492
100	0.9709	0.4849	0.5042



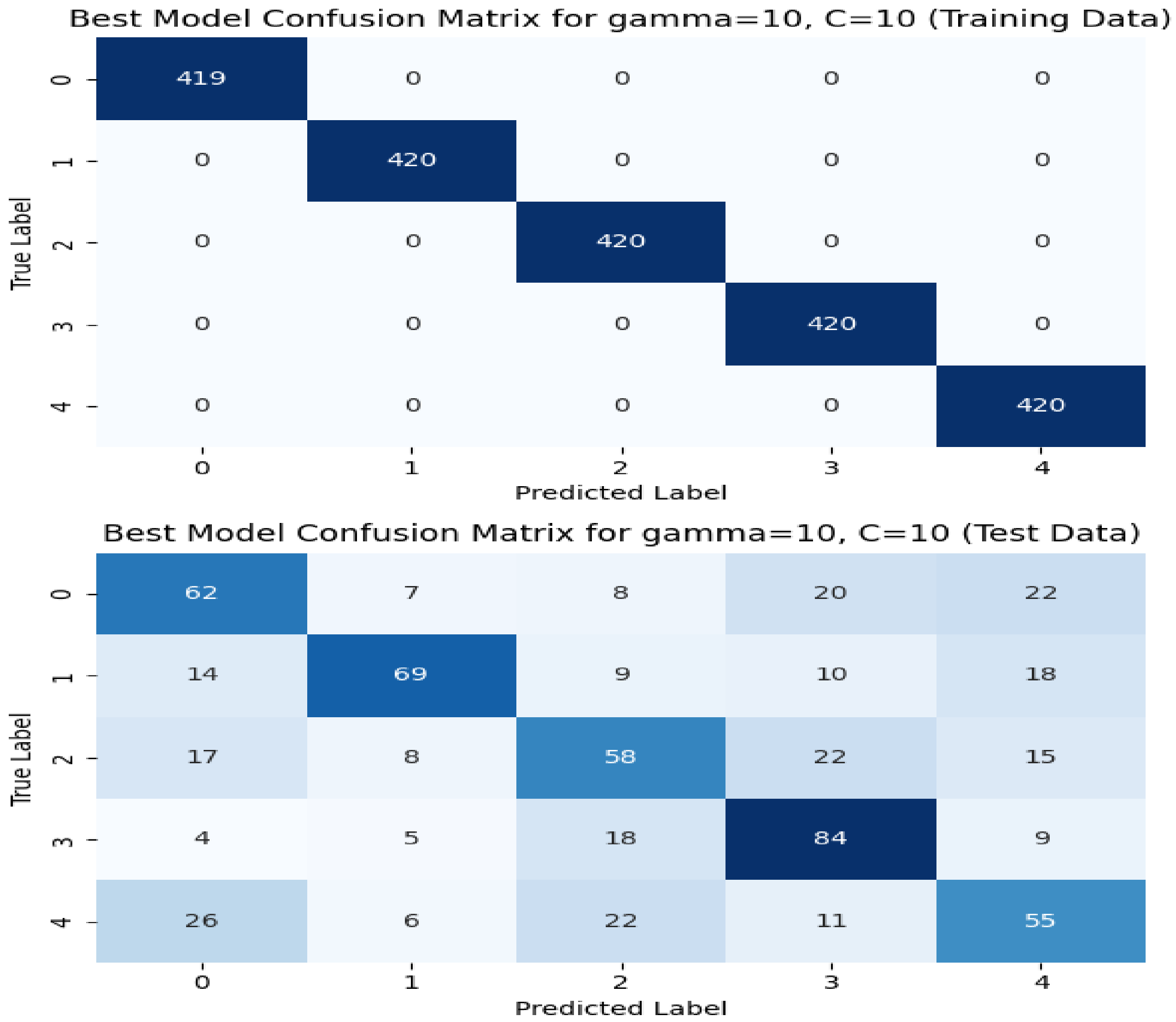
Accuracy Table for Gamma = 10.0

Gamma = 10.0	C	Train Accuracy	Validation Accuracy	Test Accuracy
	1	0.9443	0.5117	0.5626
	10	1.0000	0.5217	0.5476
	100	1.0000	0.5217	0.5476

We choose the best performing model based on the model performance on the validation date over all degree of polynomial and C values. from above Table we can see that

The best gamma is: 10
The best C value is: 10
Validation Accuracy: 0.54
Test Accuracy: 0.52
Train Accuracy: 1.0

Table: 3.5 Confusion matrices for training data and test data of the best performing model



Best Model Confusion Matrix for gamma=10, C=10 (Training Data)

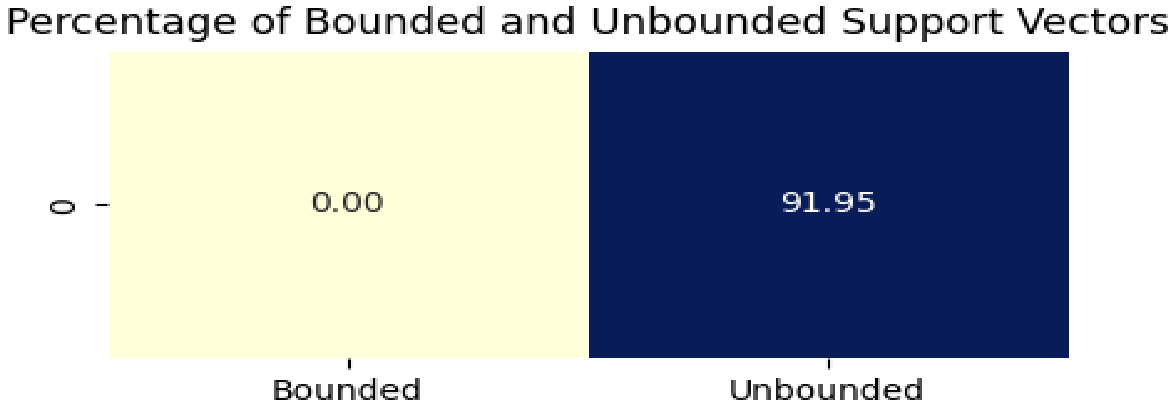
True Label \ Predicted Label	0	1	2	3	4
0	419	0	0	0	0
1	0	420	0	0	0
2	0	0	420	0	0
3	0	0	0	420	0
4	0	0	0	0	420

Best Model Confusion Matrix for gamma=10, C=10 (Test Data)

True Label \ Predicted Label	0	1	2	3	4
0	62	7	8	20	22
1	14	69	9	10	18
2	17	8	58	22	15
3	4	5	18	84	9
4	26	6	22	11	55

Table-3.6 A table of percentage of bounded and unbounded support vectors out of the

training examples, for the best performing model



Observations:

- 91.95% of Support Vectors are Unbounded: This indicates that most of the support vectors are unbounded, meaning they lie on or within the decision boundary's margin rather than directly influencing the shape of the margin.
- 0% Bounded Support Vectors: With no bounded support vectors, it appears that the regularization parameter C is allowing flexibility within the margin.

Exercise-4: PCA (Principal Component Analysis) for Dataset 3

Principal Component Analysis (PCA) is a dimensionality reduction technique that transforms the original data into a new coordinate system, where the greatest variance by any projection lies on the first coordinate (principal component), the second greatest variance on the second coordinate, and so on. This process helps in reducing the dimensionality of the data while retaining as much variance as possible.

Dataset Characteristics:

- **Original Dimensions:** 35
- **Cumulative Variance Explained for Reduced Dimensions:**
  - **80% Cumulative Variance:** Suitable reduced dimension L = 18
  - **90% Cumulative Variance:** Suitable reduced dimension L = 26
  - **95% Cumulative Variance:** Suitable reduced dimension L = 30

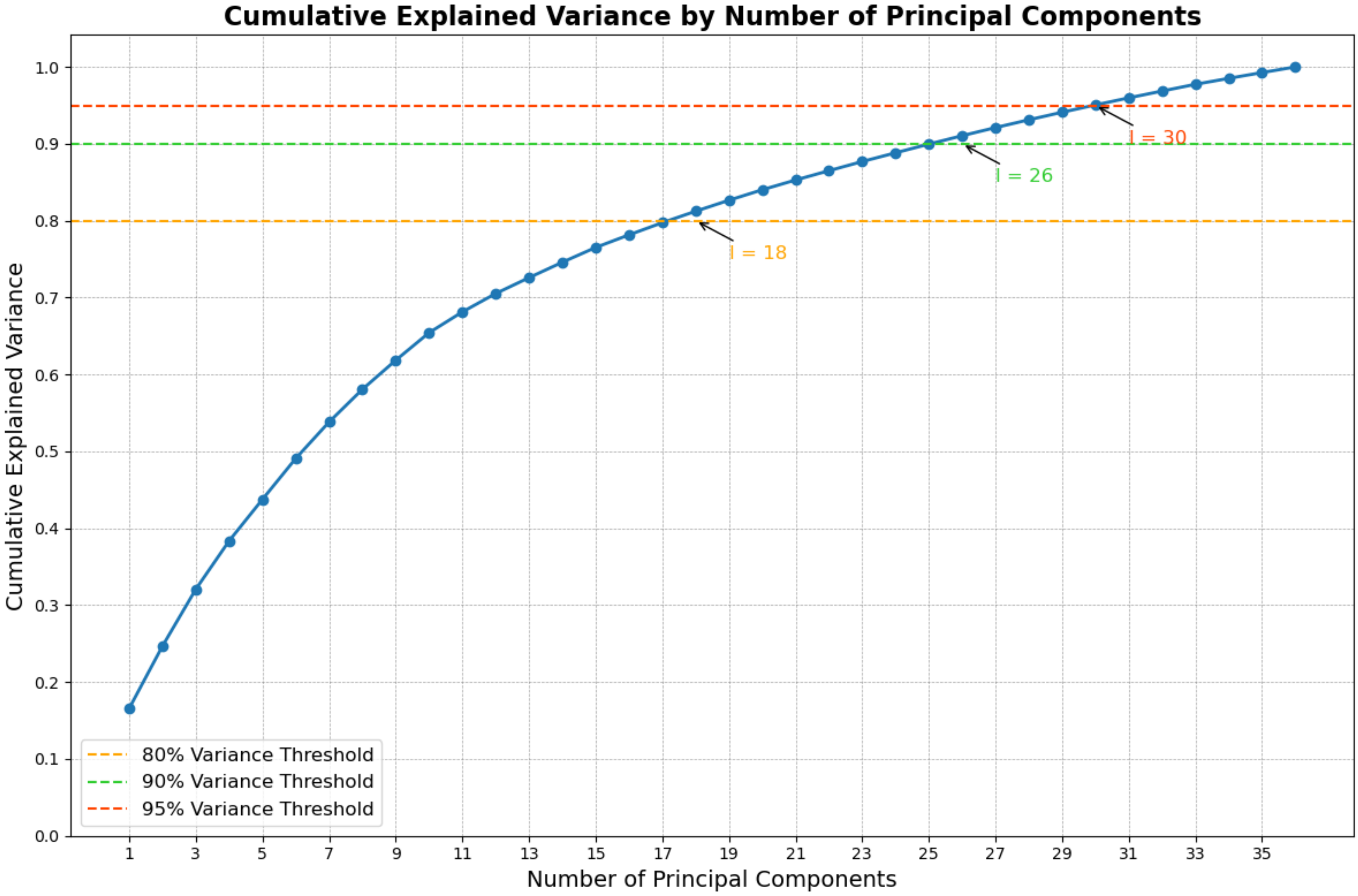


Fig-4.1

Analysis

1. **Cumulative Variance:**
   - **80% Variance:** Reducing the dimensions to 18 retains 80% of the original variance. This is suitable for applications where a moderate amount of information loss is acceptable, often used for visualization or preliminary analysis.
   - **90% Variance:** For more detailed analysis, maintaining 90% of the variance is achieved by reducing the dimensions to 26. This strikes a balance between dimensionality reduction and retaining significant data characteristics.

    - **95% Variance:** To capture a higher amount of information with minimal loss, 30 dimensions are needed to retain 95% of the variance. This is ideal for cases requiring more detailed insights without overwhelming complexity.

2. **Dimensionality Reduction Decision:**
    - The choice of reduced dimension lll depends on the specific application:
        - For **exploratory data analysis** or visualization, reducing to **18 dimensions** (80% variance) may suffice.
        - For **machine learning models** requiring higher accuracy, **26 dimensions** (90% variance) could be the better choice.
        - If maintaining maximum information is crucial, **30 dimensions** (95% variance) should be selected.

Conclusion

The PCA analysis for Dataset 3 indicates that suitable values for the reduced dimension lll are:

- **18** for 80% cumulative variance
- **26** for 90% cumulative variance
- **30** for 95% cumulative variance

This analysis allows for flexibility in choosing the dimensionality based on the balance between data representation and computational efficiency, aiding in further data processing and analysis steps.

Exercise-5: Classifier for Dataset 3 with reduced dimension representation obtained in Exercise 4, as the input to the classifier. The classification models are as follows:

a) Using GMM:

Dataset 3: Image data (Dimension of feature vector: 35) for 5 classes

This report summarizes the performance of a Gaussian Mixture Model (GMM) applied to a dataset with results obtained from training and testing phases. The confusion matrices provide insights into the classification outcomes across different classes.

Table: 5.1.1: A table of classification accuracies for the training, validation and test data using Full Covariance Matrix

Dataset	Accuracy
Train	0.74
Validation	0.51
Test	0.50

Table: 5.1.2: A table of classification accuracies for the training, validation and test data using Diagonal Covariance Matrix

Dataset	Accuracy
Train	0.56
Validation	0.47
Test	0.48

Table: 5.1.3: Confusion matrices for training data and test data

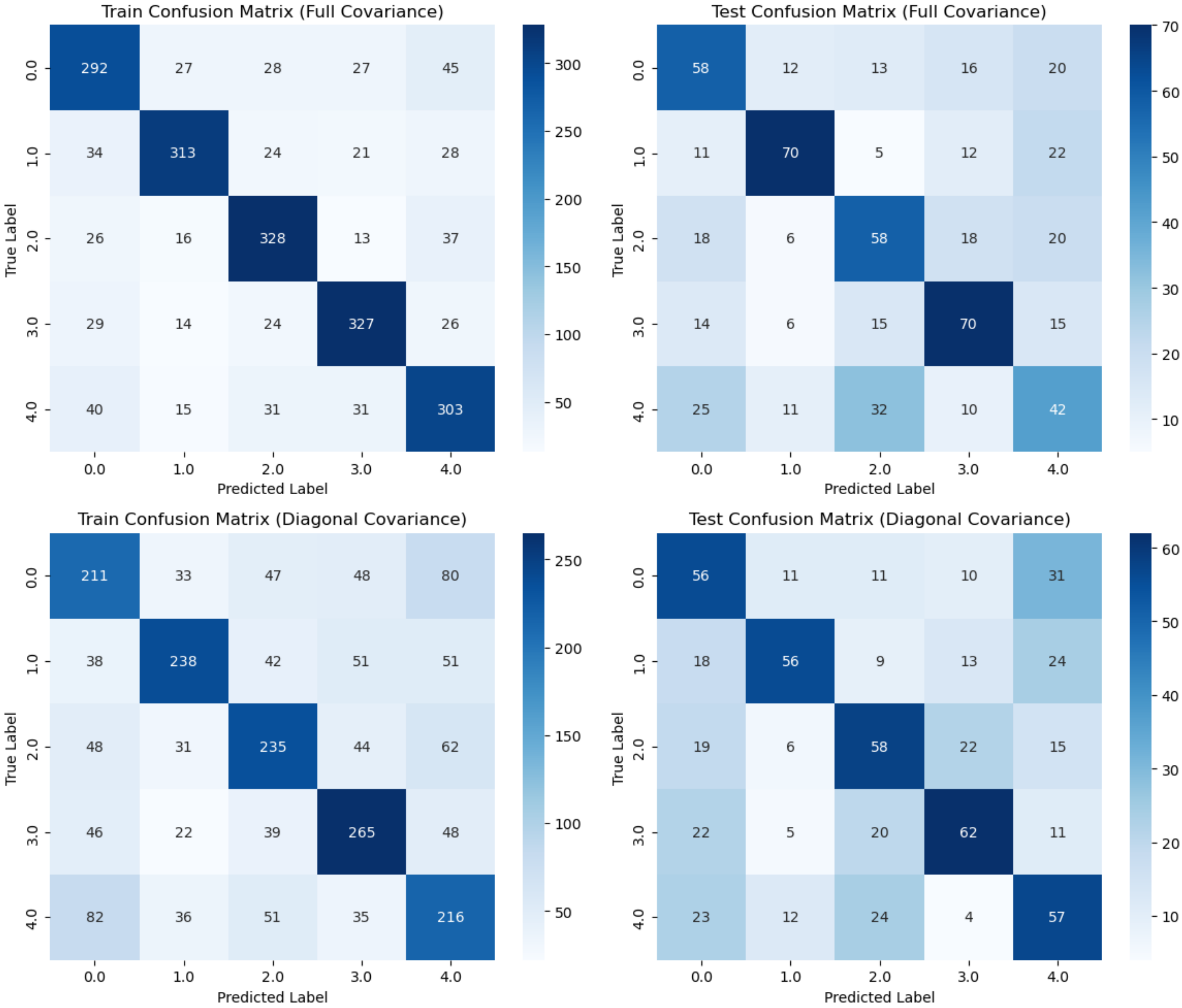


Fig-5.1.1

Analysis of Results:

1. **Training vs. Test Accuracy:**
    - The first model achieved a higher training accuracy (0.74) compared to the second model (0.56). However, the test accuracy for the first model is significantly lower (0.50) than its training accuracy, suggesting overfitting.
    - The second model's performance on both training (0.56) and test (0.48) datasets is relatively poor, indicating that the model is not learning effectively from the data.
2. **Confusion Matrices Interpretation:**
    - **First Model:**
        - The training confusion matrix indicates strong performance in most classes, especially in the third class (328 correctly classified).
        - The test confusion matrix shows significant misclassifications, particularly in classes 1 and 4, which might indicate that the model struggles to generalize from the training to the test set.
    - **Second Model:**
        - The training confusion matrix shows more evenly distributed misclassifications, with no class dominating the correct predictions.
        - The test confusion matrix further reflects this trend, with classes being misclassified across the board. This indicates a lack of robustness and potential model inadequacies.
3. **Overall Performance:**
    - The first model demonstrates better initial performance but suffers from overfitting, which can be detrimental when applied to unseen data.
    - The second model performs poorly overall, suggesting the need for further optimization, hyperparameter tuning, or potentially revisiting feature selection.

Conclusion

The results from the GMM implementation indicate challenges in achieving robust performance across the datasets. The first model, while showing higher training accuracy, is not generalizing well to the test set, highlighting the importance of balancing model complexity with the ability to generalize. The second model lacks performance entirely, suggesting the need for more refined approaches.

b) Using MLFFNN:

Dataset 3: Image data (Dimension of feature vector: 35) for 5 classes

The Multi-Layer Feedforward Neural Network (MLFFNN) was trained on the dataset for a total of 277 epochs. The results indicate the model's performance in terms of accuracy across training, validation, and test datasets.

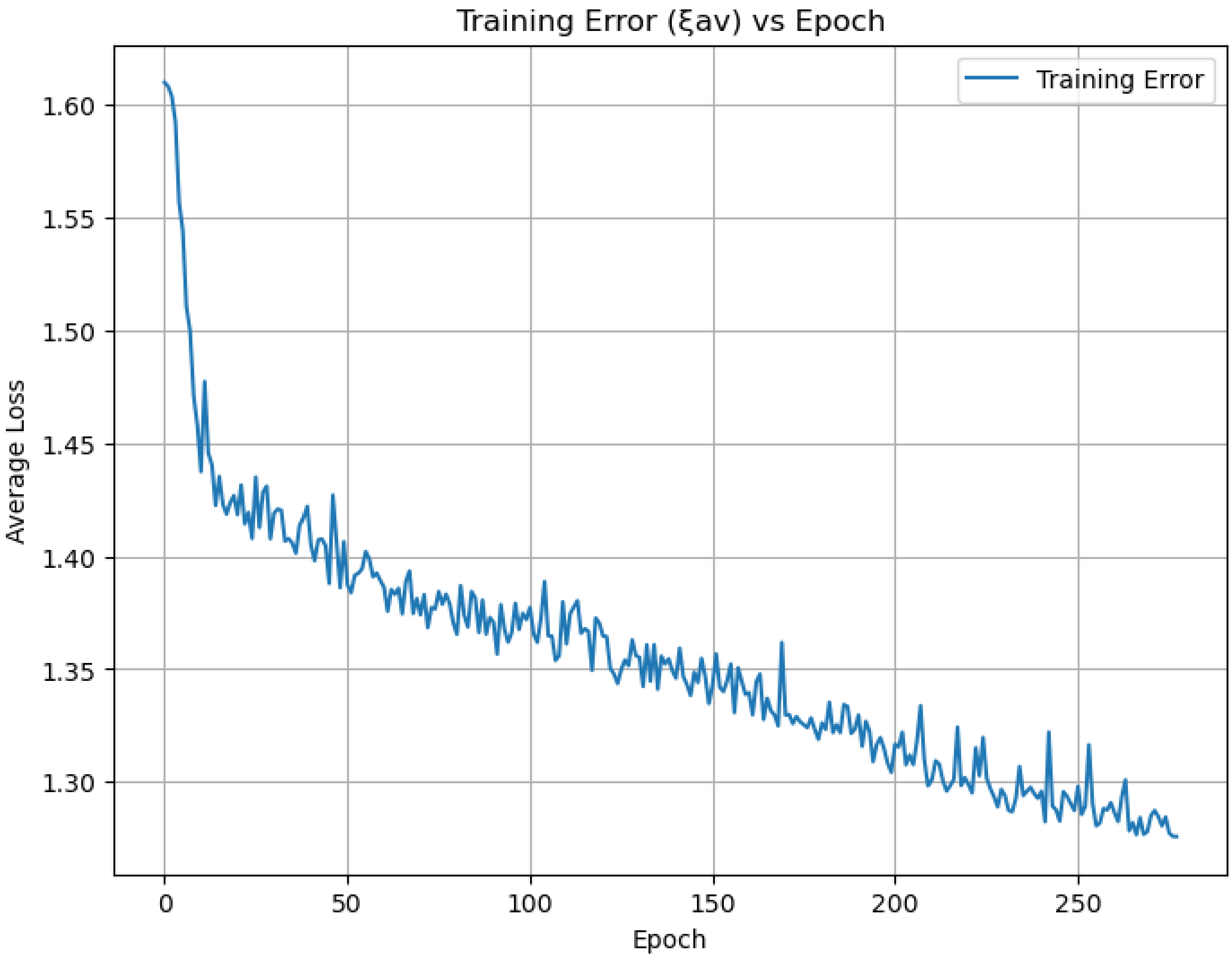
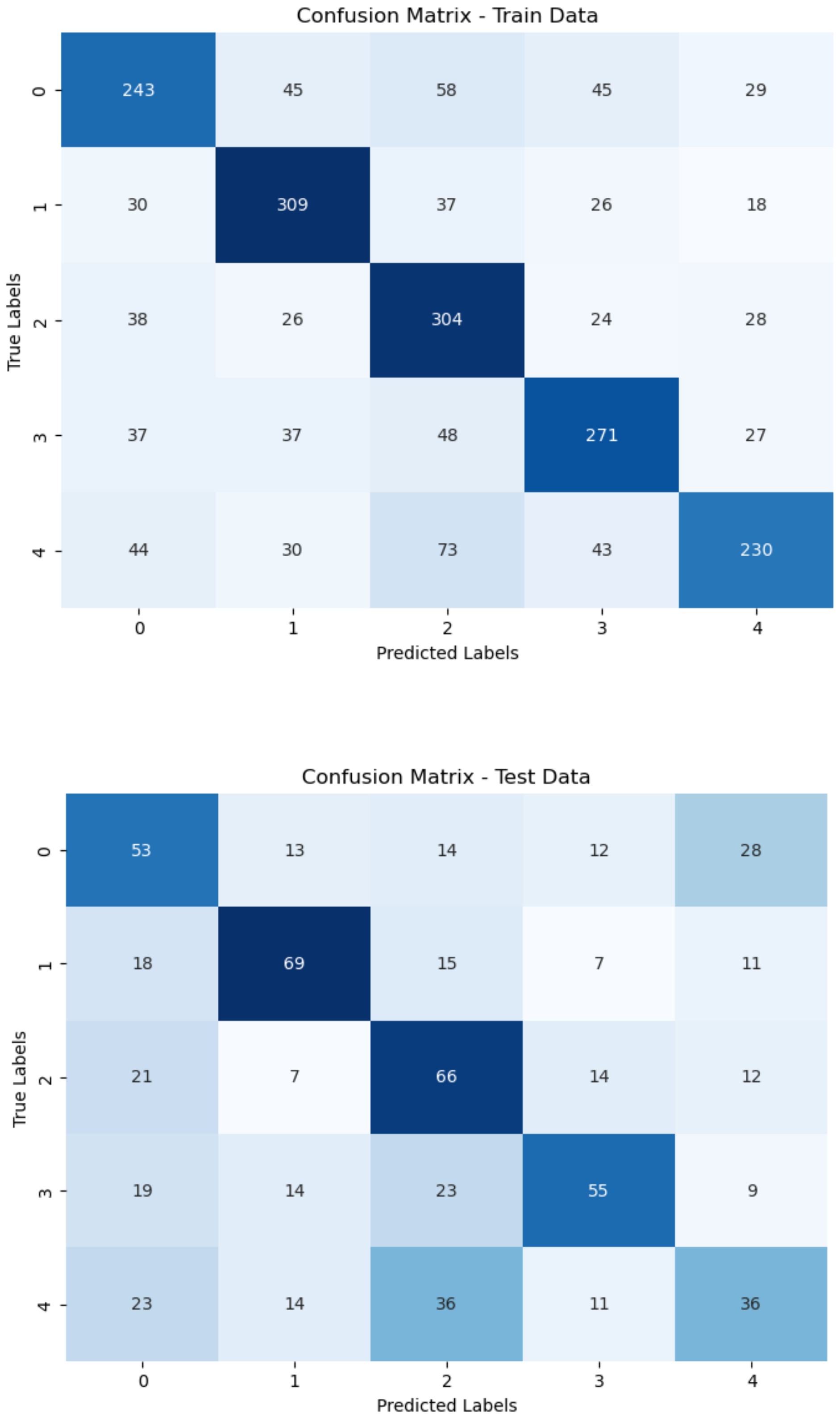


Fig-5.2.1

Table: 5.2.1: A table of classification accuracies for the training, validation and test data

Dataset	Accuracy
Train	0.6462
Validation	0.4400
Test	0.4650

Table: 5.2.2: Confusion matrices for training data and test data



Analysis of Results:

1. **Training Accuracy:**
   - The training accuracy of **64.62%** suggests that the model has learned some relevant features from the training data. However, this value is relatively moderate, indicating that there is room for improvement in model performance.

2. **Validation and Test Accuracy:**
    - The validation accuracy of **44.00%** and test accuracy of **46.50%** are significantly lower than the training accuracy. This gap indicates potential overfitting, where the model performs well on the training data but struggles to generalize to unseen data.
    - The test accuracy being slightly higher than the validation accuracy is not typical and may suggest that the test dataset has characteristics more similar to the training data than the validation set, or that the model has not sufficiently learned the underlying patterns.
3. **Convergence:**
    - The model converged after **277 epochs**, which is a substantial number of epochs. While this indicates that the model continued to learn for a considerable time, the lack of improvement in validation and test accuracy suggests that it may have plateaued without achieving robust performance.

Conclusion:

The MLFFNN demonstrates a moderate level of training accuracy but struggles to generalize effectively, as seen in the validation and test results. By implementing the recommendations outlined above, there is potential for significant improvement in model performance on unseen data. Further analysis and adjustments will be crucial to achieve better accuracy and reliability in predictions.

c) Using SVM:

Dataset 3: Image data (Dimension of feature vector: 35) for 5 classes

This report summarizes the performance of two Support Vector Machine (SVM) models using different kernels—Polynomial and Gaussian. The results include training and test accuracies, along with an analysis of support vectors for both models.

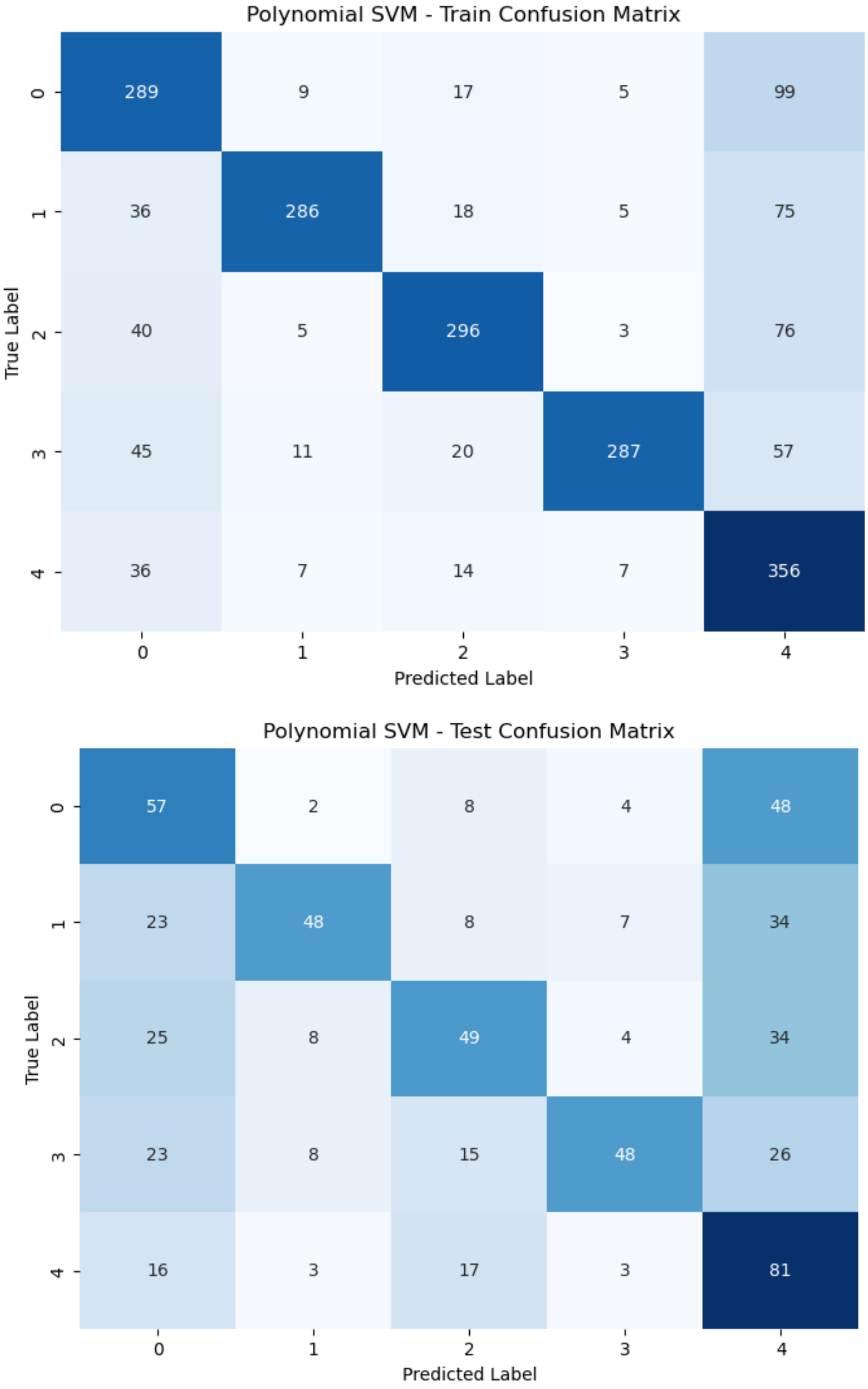
Table: 5.3.1: A table of classification accuracies for the training, validation and test data using Polynomial Kernel SVM

Dataset	Accuracy
Train	0.72
Validation	0.44
Test	0.47

Table: 5.3.2: A table of classification accuracies for the training, validation and test data using Gaussian Kernel SVM

Dataset	Accuracy
Train	0.85
Validation	0.53
Test	0.55

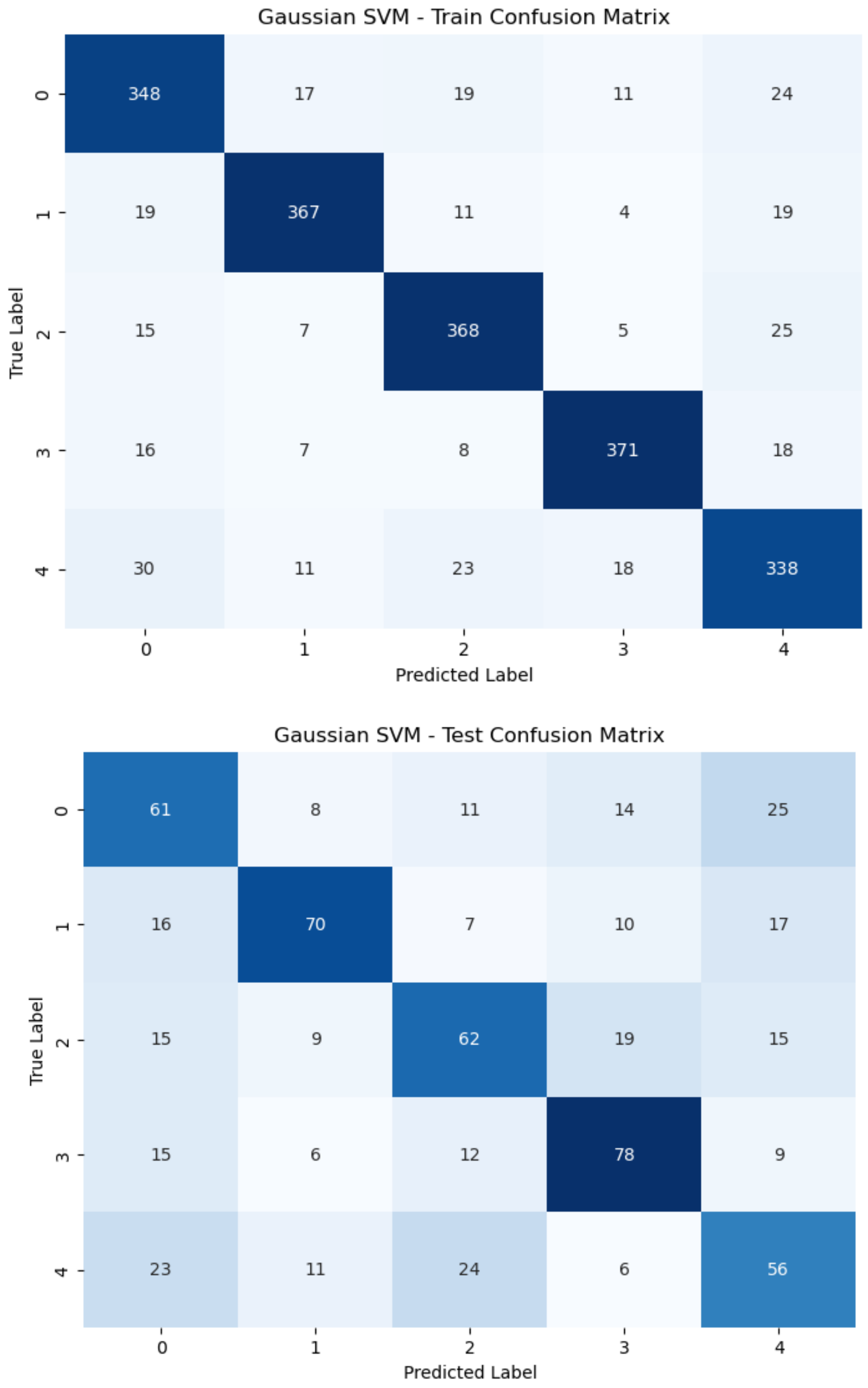
Table: 5.3.3: Confusion matrices for training data and test data using Polynomial Kernel SVM



Polynomial SVM - Train Confusion Matrix

Polynomial SVM - Test Confusion Matrix

Table: 5.3.4: Confusion matrices for training data and test data using Gaussian Kernel SVM



Gaussian SVM - Train Confusion Matrix

	0	1	2	3	4
0	348	17	19	11	24
1	19	367	11	4	19
2	15	7	368	5	25
3	16	7	8	371	18
4	30	11	23	18	338

Gaussian SVM - Test Confusion Matrix

	0	1	2	3	4
0	61	8	11	14	25
1	16	70	7	10	17
2	15	9	62	19	15
3	15	6	12	78	9
4	23	11	24	6	56

Model Results

1. Polynomial Kernel SVM:

- **Training Accuracy:** 0.72 (72%)
- **Test Accuracy:** 0.47 (47%)

Support Vector Analysis:

- **Total Support Vectors:** 1841

- **Bounded Support Vectors:** 1539 (83.60%)
- **Unbounded Support Vectors:** 302 (16.40%)

Parameters:

- Degree: 3
- C (Regularization Parameter): 1

---

2. Gaussian Kernel SVM:

- **Training Accuracy:** 0.85 (85%)
- **Test Accuracy:** 0.55 (55%)

Support Vector Analysis:

- **Total Support Vectors:** 1961
- **Bounded Support Vectors:** 1654 (84.34%)
- **Unbounded Support Vectors:** 307 (15.66%)

Parameters:

- Gamma: 10
- C (Regularization Parameter): 10

Overall Results Comparison

Model	Train Accuracy	Test Accuracy
Polynomial SVM	0.721296	0.472454
Gaussian SVM	0.853740	0.545910

Analysis of Results

1. **Training and Test Accuracy:**
   - The **Gaussian Kernel SVM** outperforms the **Polynomial Kernel SVM** in both training (85% vs. 72%) and test accuracy (55% vs. 47%). This indicates that the Gaussian kernel is more effective at capturing the data distribution and generalizing to unseen data.
   - The significant difference between training and test accuracies for both models suggests that while the models can learn patterns from the training data, there may be challenges in generalization, particularly for the Polynomial kernel.
2. **Support Vector Analysis:**
   - Both models have a substantial number of total support vectors, with the Gaussian kernel slightly leading in this regard (1961 vs. 1841).
   - The percentage of bounded support vectors is similar for both models, indicating that the majority of support vectors are contributing to defining the decision boundary. However, the higher number of support vectors in the Gaussian kernel may imply a more complex decision boundary.
3. **Model Complexity:**
   - The choice of kernel significantly impacts the model's complexity and performance. The Polynomial kernel, despite having lower training accuracy, may struggle with the complexity of the data, leading to lower performance on the test set.
   - The Gaussian kernel, with its radial basis function, can better capture non-linear relationships, which could explain its superior performance.

Conclusion

The Gaussian kernel SVM provides better training and test accuracy compared to the Polynomial kernel SVM, indicating it is more effective for this dataset. Both models exhibit a noticeable gap between training and test accuracies, highlighting the need for further analysis and potential adjustments, such as hyperparameter tuning (especially for γ and C), feature selection, or data preprocessing techniques.