Programming Assignment - No. 1

Report Submitted in Partial Fulfilment of the Requirements for the Course

CS401 Introduction to Machine Learning

Submitted by

Group number 05

Bryson Gracias (Roll No. 20CSE1008)

Shakshi Chourase (Roll No. 20CSE1032)

Department of Computer Science and Engineering

National Institute of Technology, Goa

September 23, 2023

Contents

1 Introduction

This project aims to implement the four types of classifiers taught as a part of the coursework in the course CS401 - Introduction to Machine Learning. The four types of classifiers viz. Nearest Neighbour Classifier, K-Nearest Neighbour Classifier, Reference Template-based Classifier and Bayesian Classifier are trained on three separate datasets.

The first dataset consists of three linearly separable classes. The second dataset consists of two nonlinearly separable classes. The third dataset consists of three overlapping classes.

The results for each classifier are plotted and the observations and inferences are summarised in this report.

The code for the entire project has been implemented in Python.

Linearly Separable Classes

Our dataset consists of three linearly separable classes. We have labelled the features as Feature 1 and Feature 2, while the class labels corresponding to Class 0 is 0, Class 1 is 1 and Class 2 is 2. The training dataset along with the mean for each class is shown in Figure 1. As observed from Figure 1, the mean for class 0 is (0.06, -0.06), class 1 is (9.35, 9.56) and class 2 is (10.06, -6.88). We now try to apply the nearest neighbour classifier on this dataset to observe the decision boundaries and accuracy of the classifier.

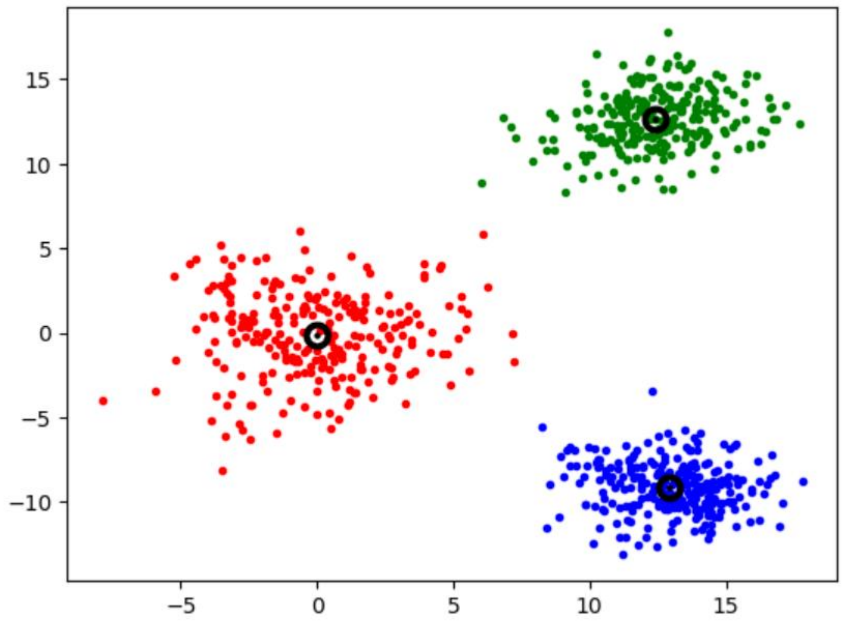


Figure 1: Training data of the linearly separable classes along with the means of each class

1.1 Nearest Neighbour Classifier

We have implemented the Nearest Neighbour (NN) Classifier in Python using the KNeigboursClassifier model from sklearn. Since the distance metric is calculated with respect to every point, we have set the value of K to be 1 for this case. The performance of the classifier on the data was measured using the following measures. Their values for the NN classifier are listed below:

Accuracy on Test Set: 100%

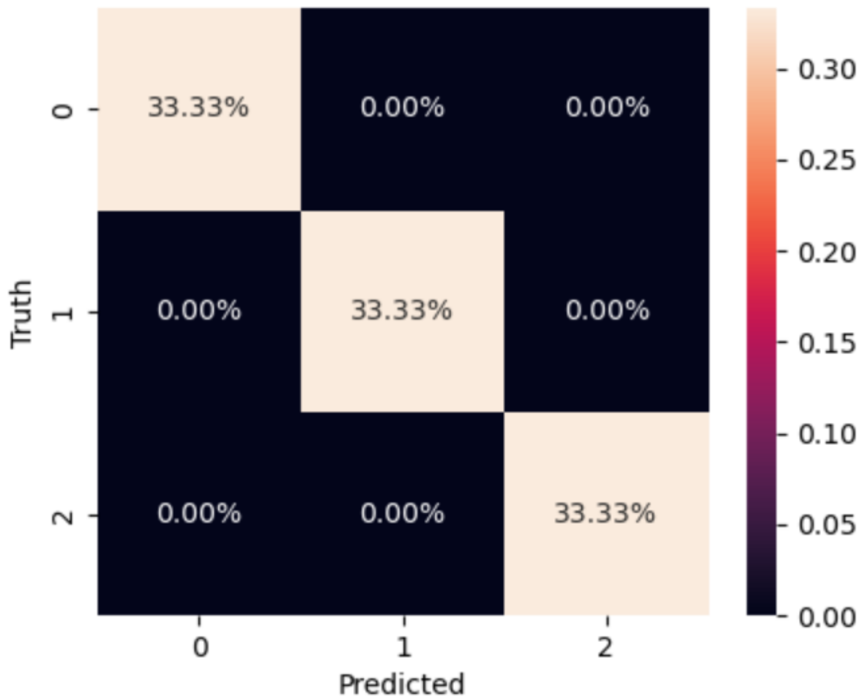
Accuracy on Validation Set: 100%

Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100% **Confusion Matrix:**

Table 1: Confusion Matrix for NN classifier implemented on Linearly Separable Classes



The decision region plot for all the classes together with the training data superimposed is shown in Figure 2, while the decision region plot for all the classes together with the testing data superimposed is shown in Figure 3.

As observed from the figures, it is evident that none of the samples are classified incorrectly, which thus signifies the 100% obtained across all performance measures.

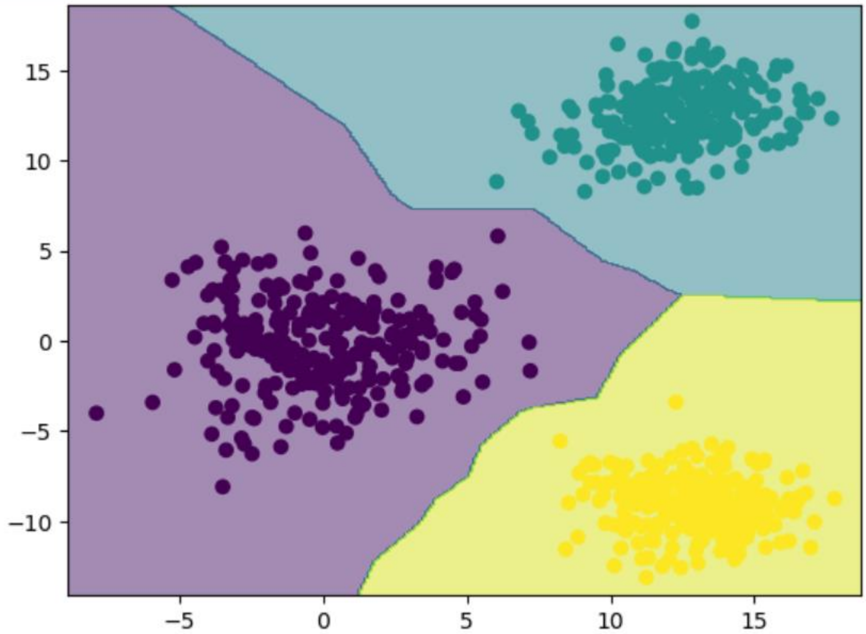


Figure 2: Decision region plot for all the classes with the training data superimpose

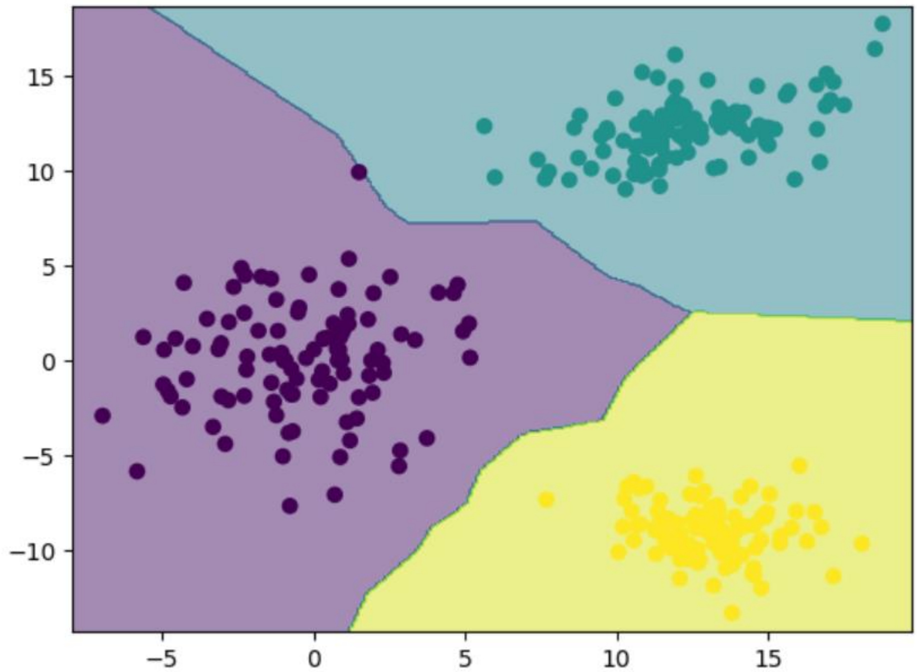


Figure 3: Decision region plot for all the classes with the testing data superimposed

As observed from the figures above, the NN classifier classifies all samples correctly, leading to 100% accuracy across all performance measures.

1.2 K-Nearest Neighbour Classifier

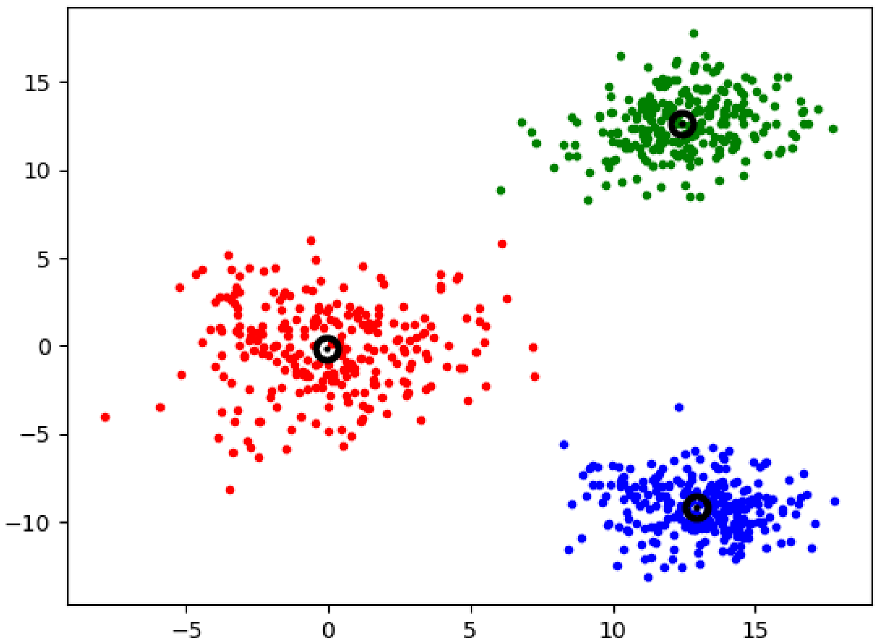


Figure 4: K-Nearest Neighbour Classifier

We have implemented the K- Nearest Neighbour Classifier in Python using the KNeigboursClassifier model from sklearn. In order to find the K value that is optimal for our case, we have fit the classifier on the validation set for increasing values of K up to half of the length of the training dataset. The plot of K values vs. accuracy of the classifier on the validation set is shown in Figure 5. As observed from Figure 5, the accuracy is maximum (100%) at

K=1. We have therefore used K=1 to train our classifier. Hence, there is no change from our previous implementation of the Nearest Neighbour Classifier. The performance of the classifier on the data was measured using the following measures. Their values for the KNN classifier are listed below:

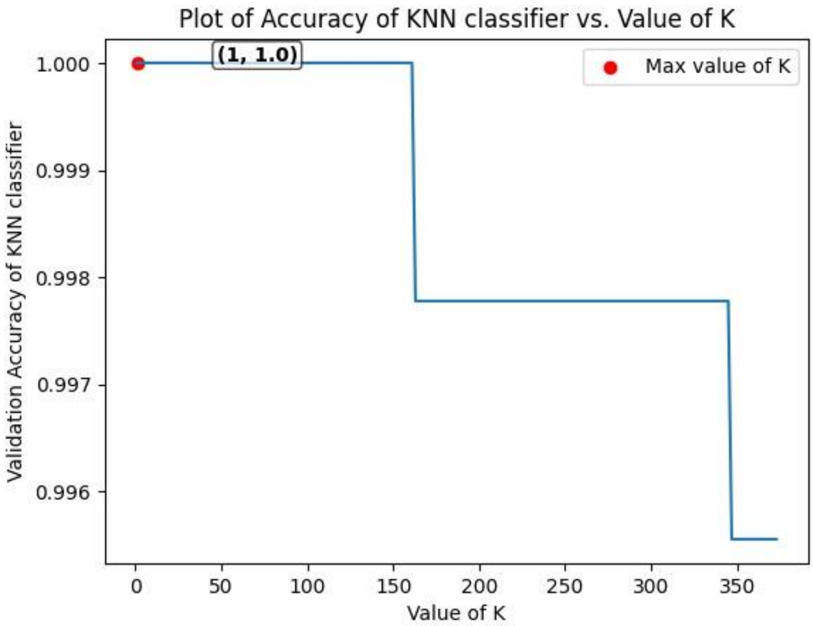


Figure 5: Plot of Accuracy of the KNN classifier on the Validation Set vs. the value of K

Accuracy on Test Set: 100%

Accuracy on Validation Set: 100%

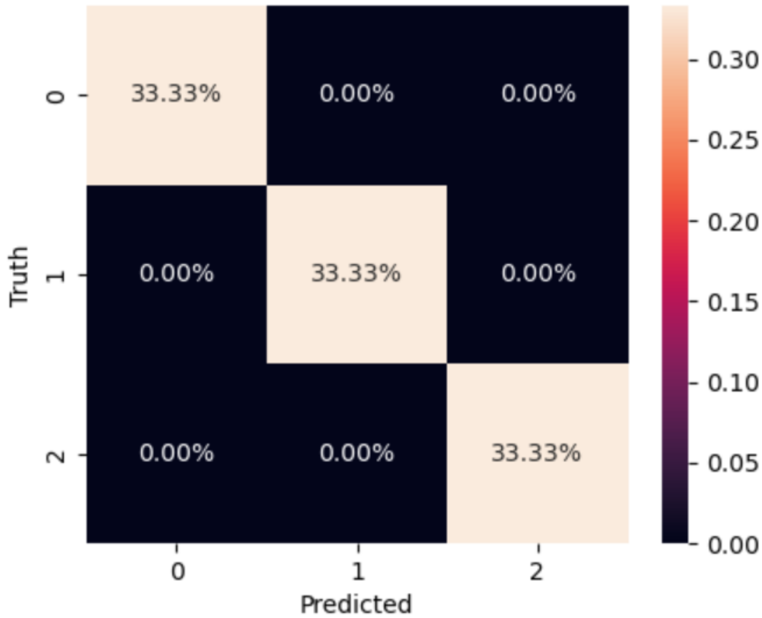
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 2: Confusion Matrix for KNN classifier implemented on Linearly Separable Classes



The decision region plot for all the classes together with the training data superimposed is shown in Figure 6, while the decision region plot for all the classes together with the testing data superimposed is shown in Figure 7.

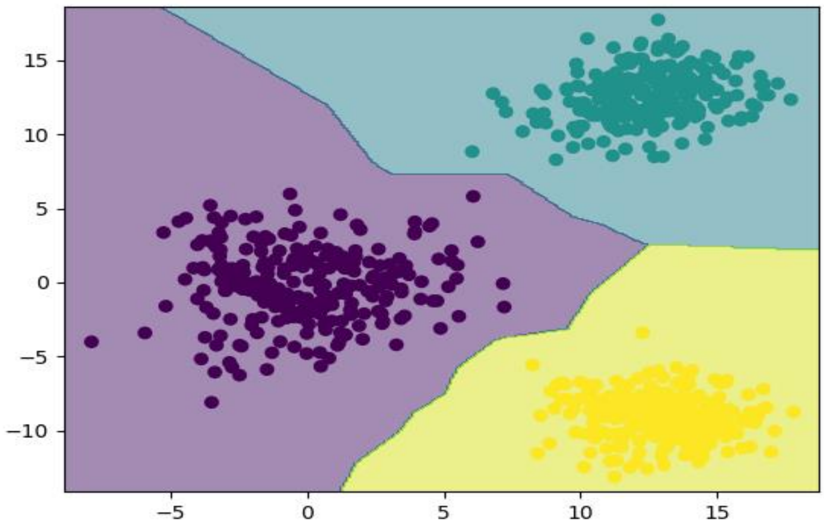


Figure 6: Decision region plot for all the classes with the training data superimposed

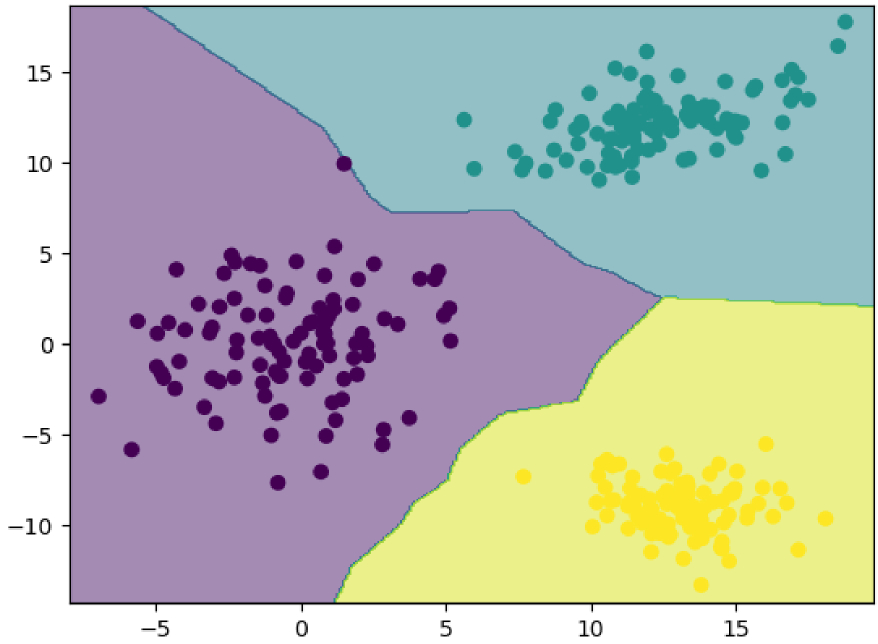


Figure 7: Decision region plot for all the classes with the testing data superimposed

As observed from the figures above, the results are the same as the Nearest Neighbour Classifier. The KNN classifier also classifies all samples correctly, leading to the 100% across all performance measures.

1.3 Reference Template-Based Classifier

We have implemented the reference template-based classifier for both sample mean and sample mean and covariance matrix-based classifier.

2.3.1 Mean Vector as Reference Template for a Class

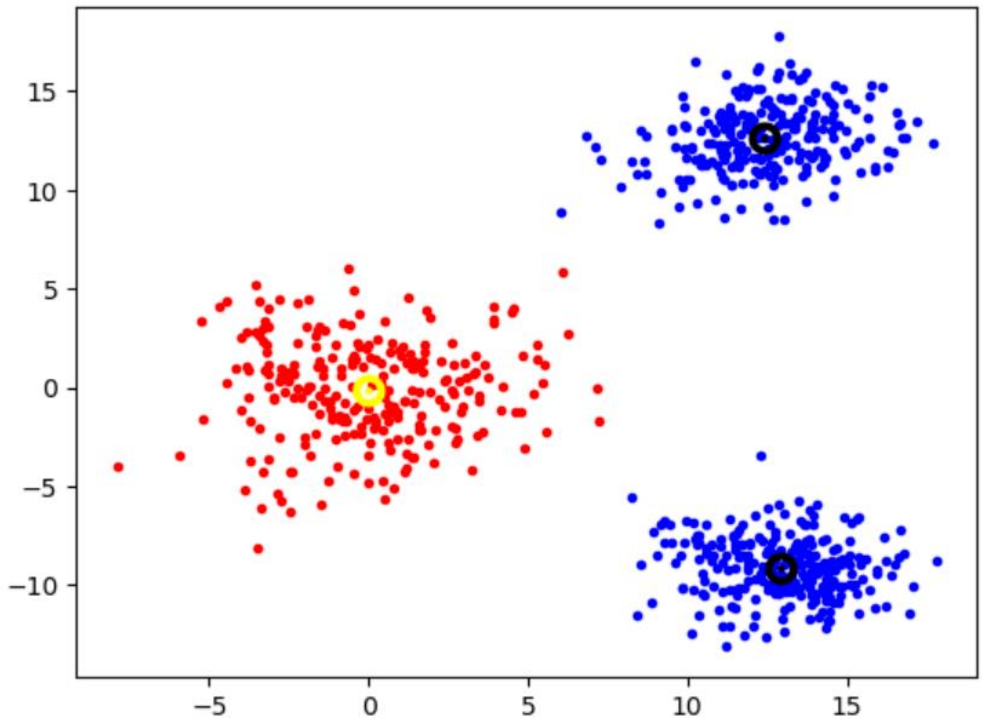


Figure 8: Mean Vector as Reference Template for a Class

We have implemented our own function for the mean vector-based classifier. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Training Set: 99.732%

Accuracy on Testing Set: 100%

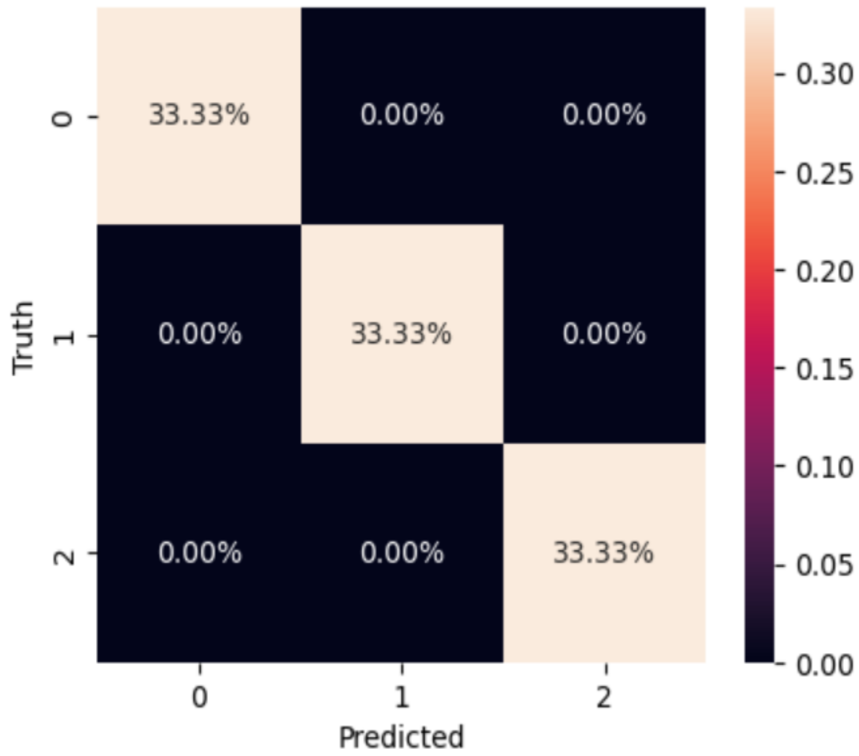
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 3: Confusion Matrix for mean vector-based classifier implemented on Linearly Separable Classes



The decision region plots for all the classes along with the training data superimposed is shown in Figure 9, while the decision region plots for all the classes along with the testing data superimposed is shown in Figure 10.

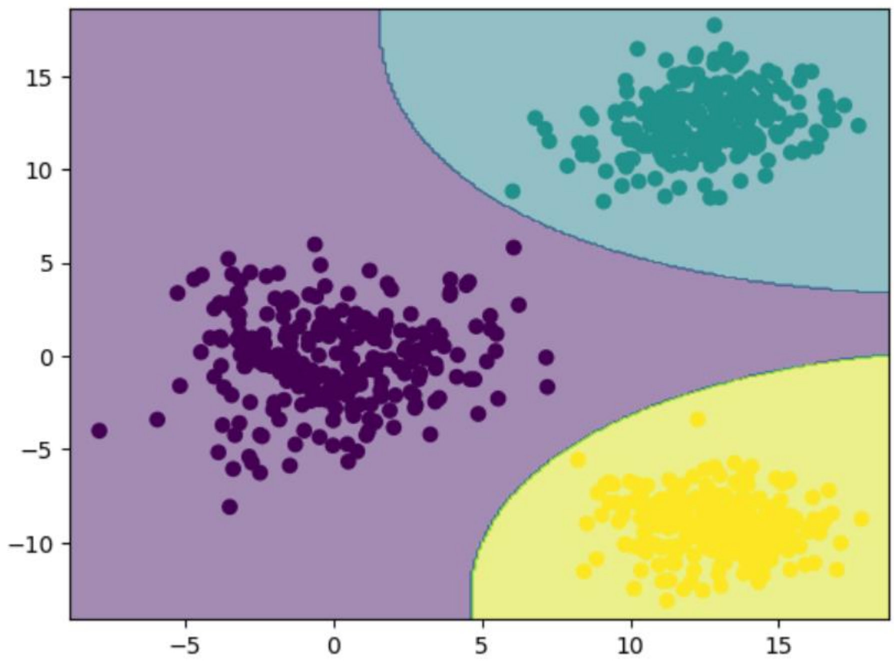


Figure 9: Decision region plot for all the classes along with the training data superimposed

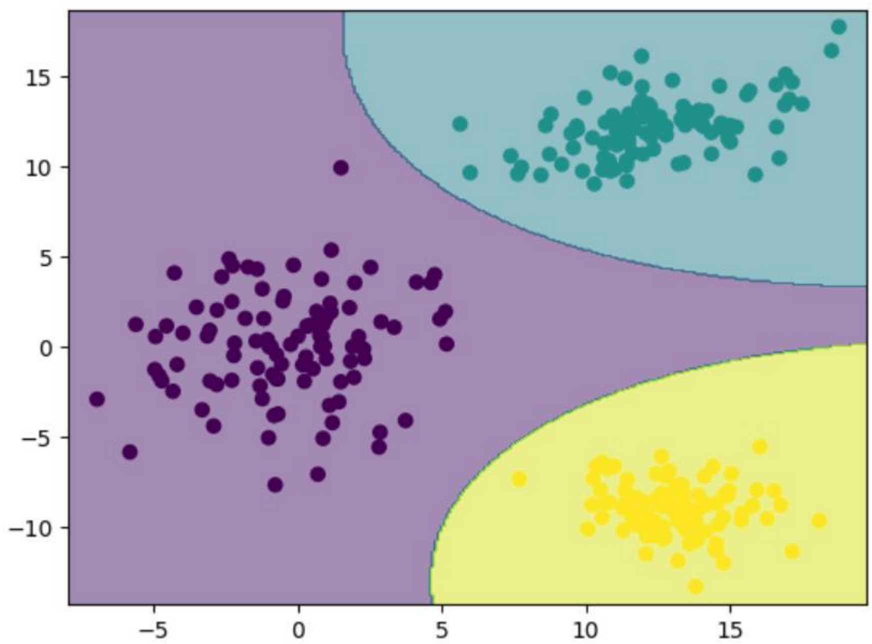


Figure 10: Decision region plot for all the classes along with the testing data superimposed

2.3.2 Mean Vector and Covariance Matrix as Reference Template for a Class

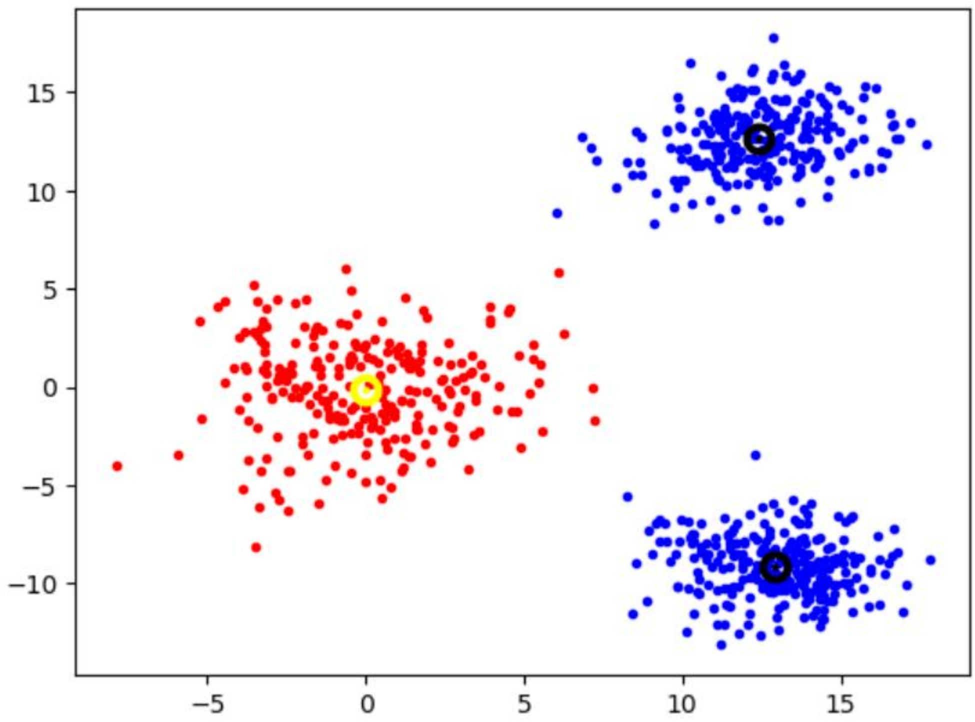


Figure 11: Mean Vector and Covariance Matrix as Reference Template for a Class

We have implemented our own function for the mean vector and covariance matrix-based classifier. The Mahalonobis distance metric was imported from the scipy library. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Testing Set: 97%

Accuracy on Validation Set: 97.111%

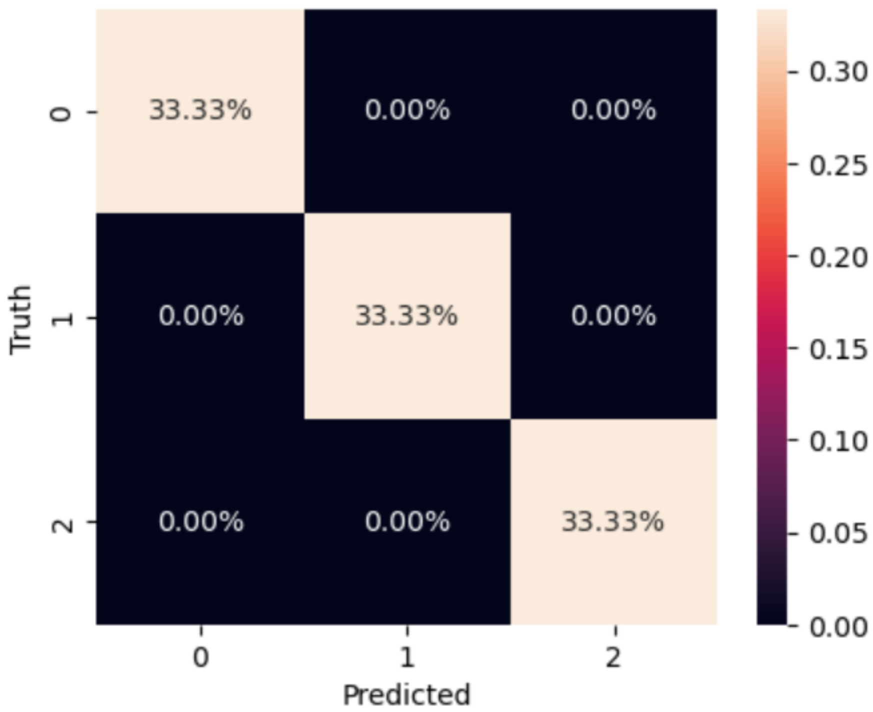
Mean Precision: 97.117%

Mean Recall: 97.00%

Mean F1-score on Testing Dataset: 96.99%

Confusion Matrix:

Table 4: Confusion Matrix for mean vector and covariance matrix-based classifier implemented on Linearly Separable Classes



The decision region plots for all the classes along with the training data superimposed is shown in Figure 12, while decision region plots for all the classes along with the training data superimposed is shown in Figure 13.

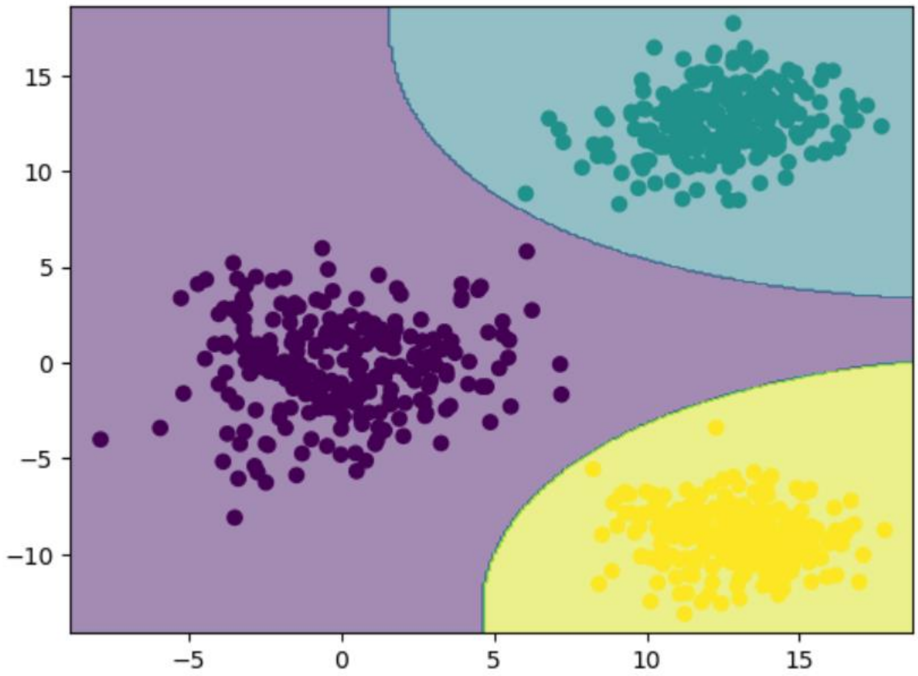


Figure 12: Decision region plot for all classes along with the training data superimposed

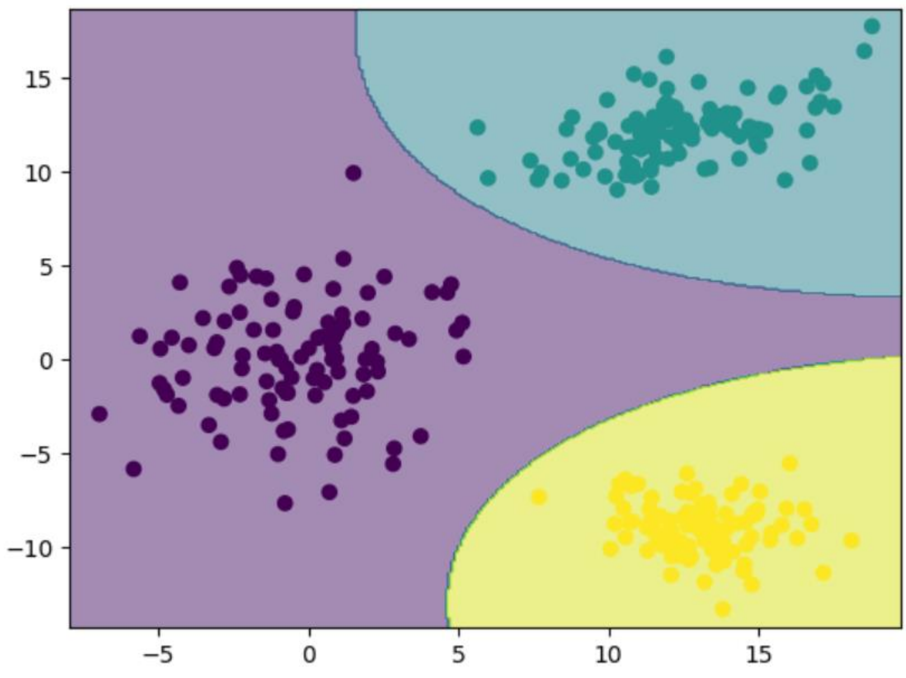


Figure 13: Decision region plot for all classes along with the testing data superimposed

As observed from the figures above, the decision boundaries formed are nonlinear.

1.4 Bayes Classifier-Unimodal Gaussian Density

Bayes Classifier was implemented for all four cases of the covariance matrix. The four cases are listed below.

2.4.1 Covariance matrix for all the classes is the same and is $\sigma^2 I$

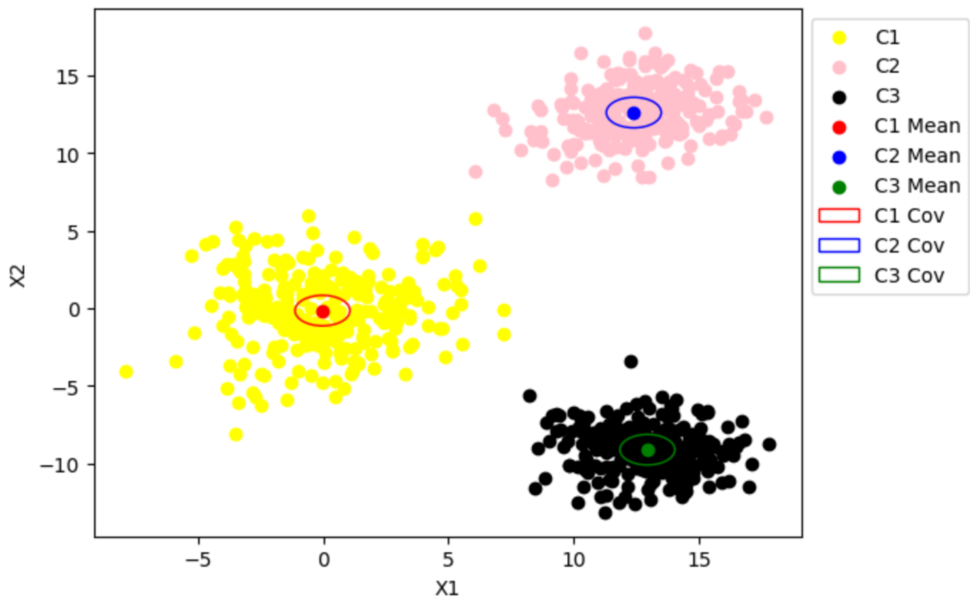


Figure 14: Covariance matrix for all the classes is the same and is σ2I

The features are independent in this case and both the features have the same variance. The Gaussian density function was implemented using the *multivariate _normal* module from scipy.stats. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Testing Set: 100%

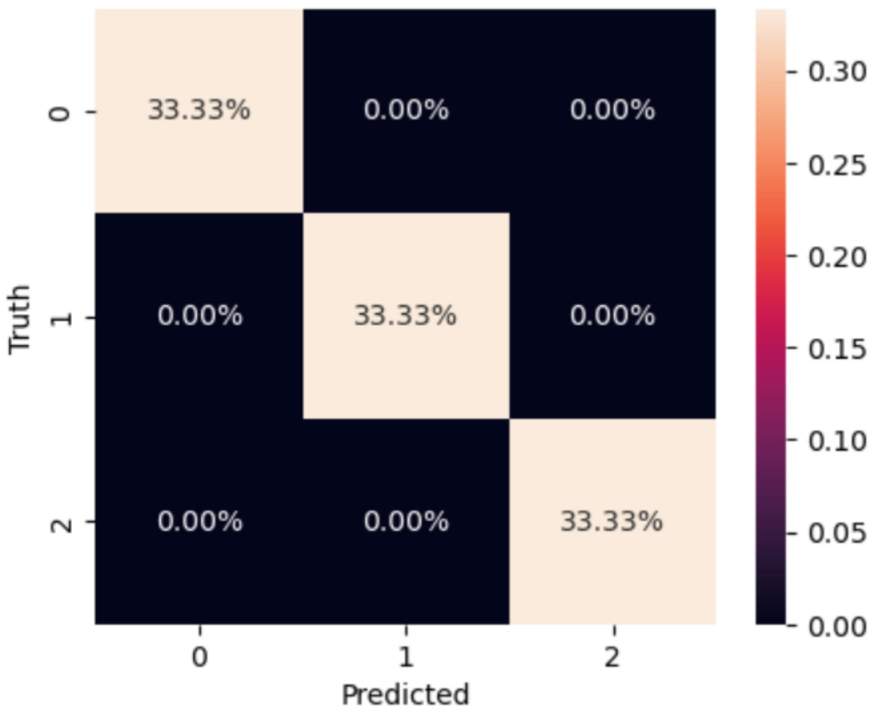
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 5: Confusion Matrix for Bayes Classifier's Case 1 implemented on Linearly Separable Classes



The decision region plots for all classes along with the training data superimposed is shown in Figure 15, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 16.

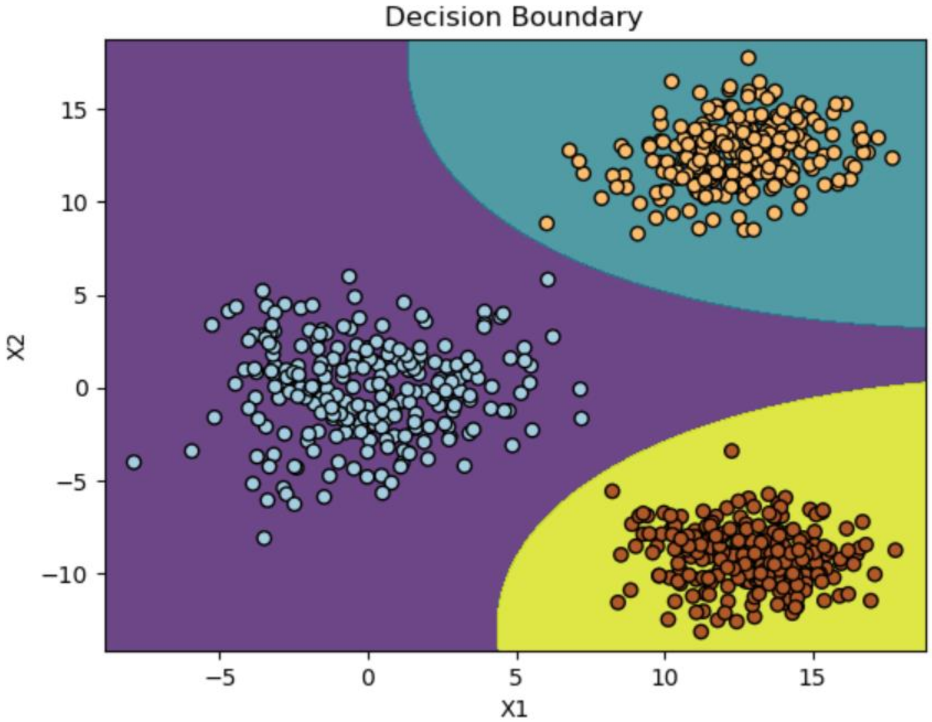


Figure 15: Decision region plot for all classes along with the training data superimposed as obtained by the Bayes Classifier's Case 1

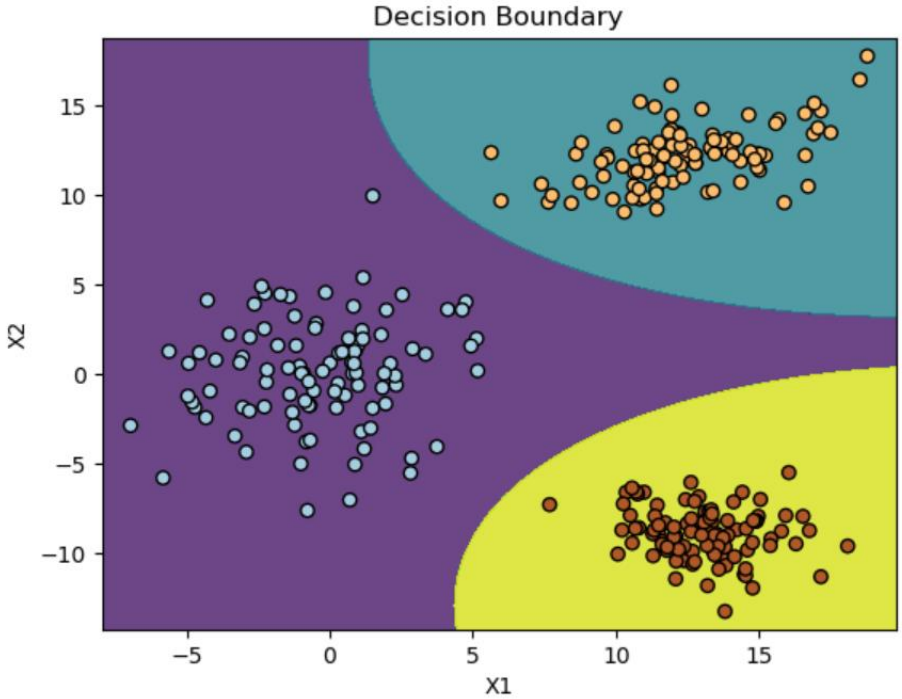


Figure 16: Decision region plot for all classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 1

As observed from the figures above, the decision boundaries formed are linear.

2.4.2 Full covariance matrix for all the classes and is same for all the classes - Part 1

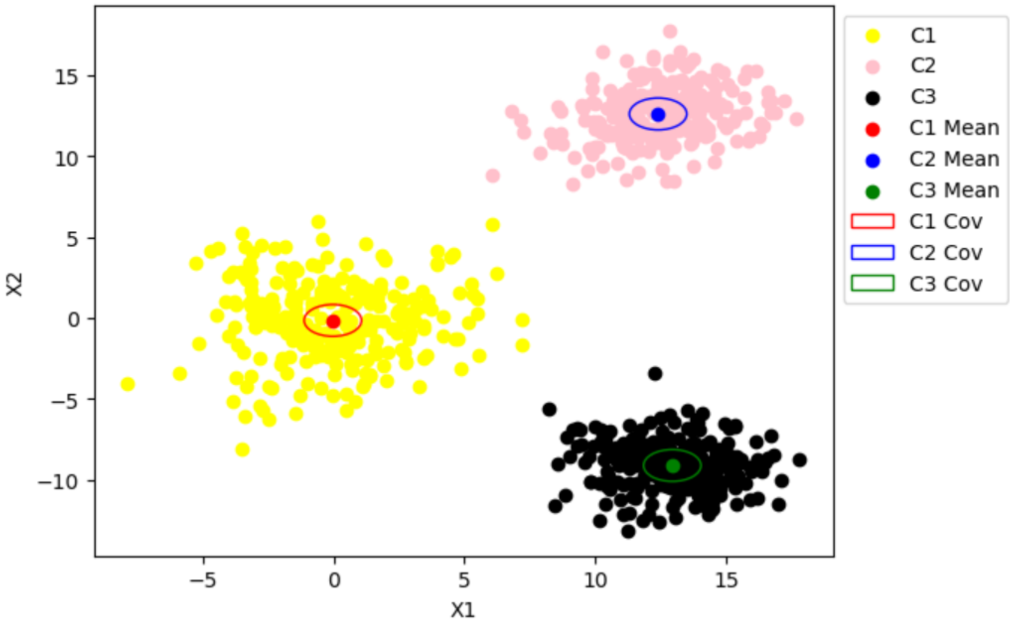


Figure 17: Full covariance matrix for all the classes and is same for all the classes - Part 1

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by taking the average of covariance matrices of all the classes. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Testing Set: 100%

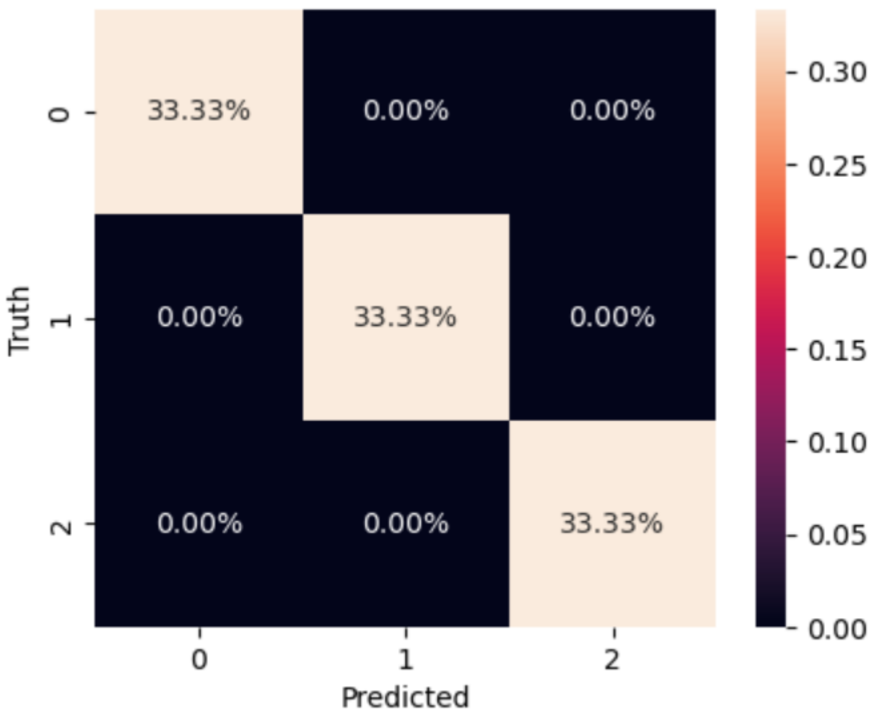
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 6: Confusion Matrix for Bayes Classifier's Case 2 Part 1 implemented on Linearly Separable Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 18, while the decision region plot for all classes along with the testing data superimposed in shown in Figure 19. As seen from the figures, the decision boundaries formed by the classifier are linear. The classifier classifies all the examples in the testing set accurately, as observed from the confusion matrix in Table 6.

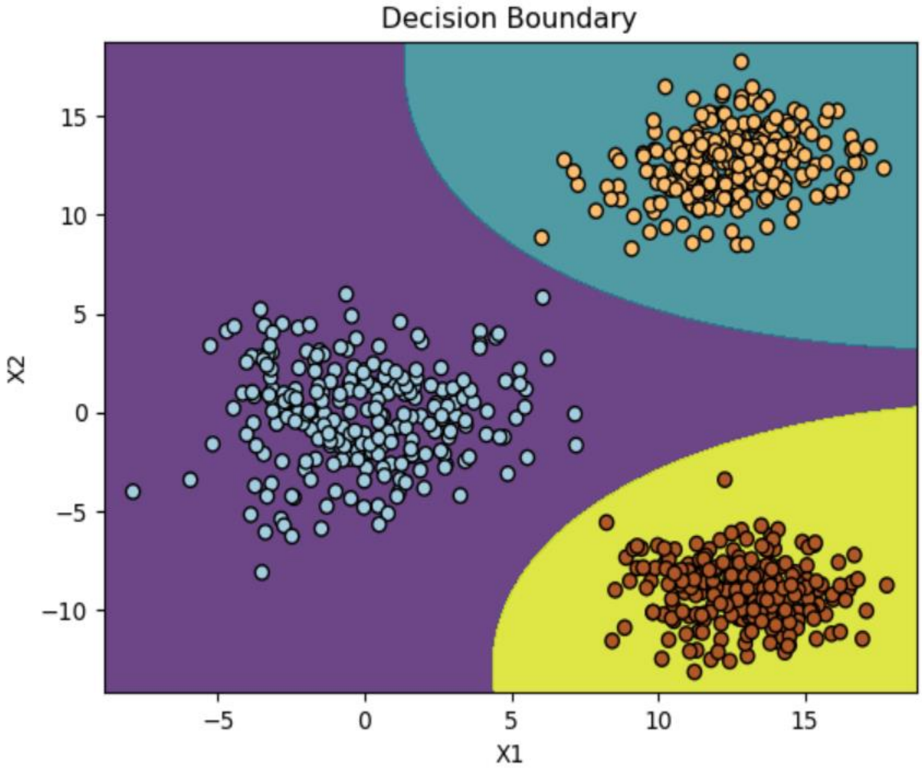


Figure 18: Decision region plot for all classes along with the training data superimposed as obtained by the Bayes Classifier's Case 2 Part 1

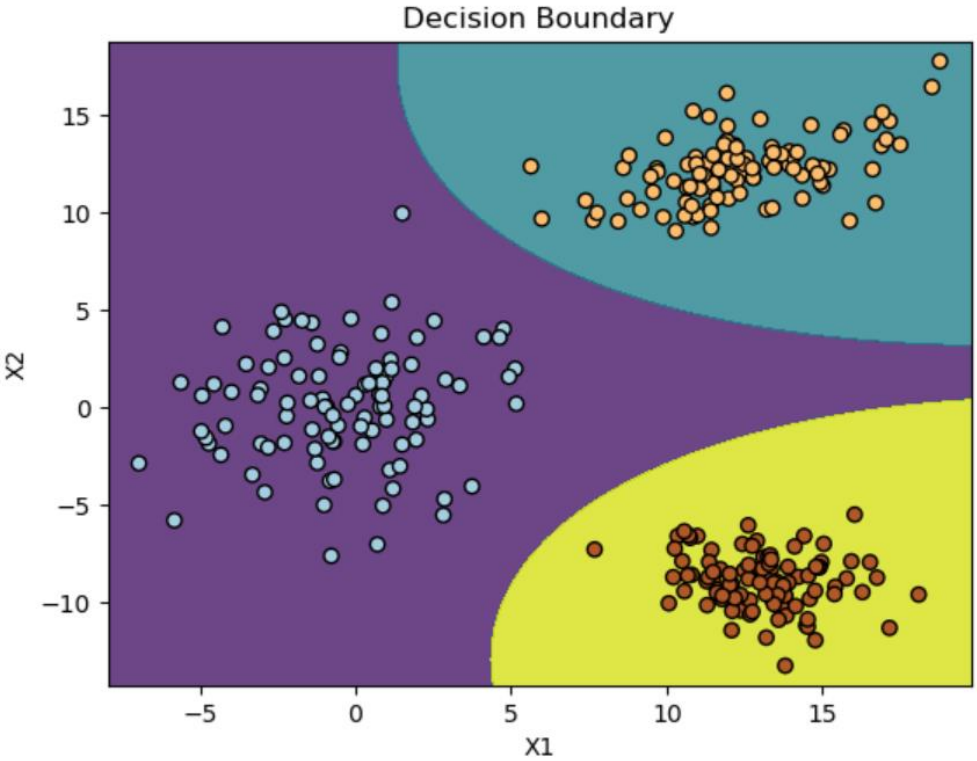


Figure 19: Decision region plot for all classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 2 Part 1

2.4.3 Full covariance matrix for all the classes and is same for all the classes - Part 2

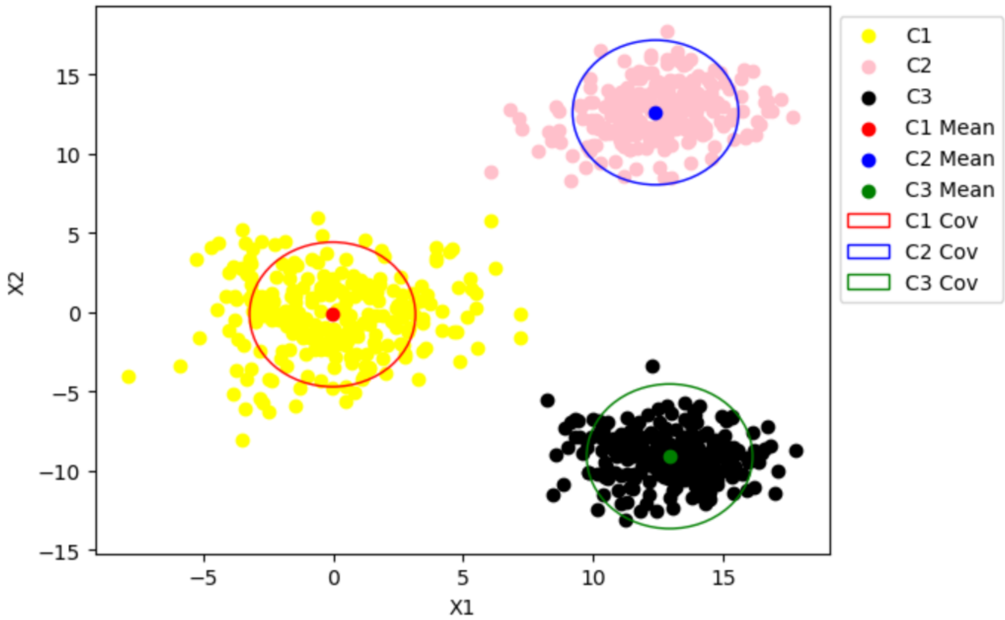


Figure 20: Full covariance matrix for all the classes and is same for all the classes - Part 2

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by computing the covariance matrix of training data of all the classes combined. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Testing Set: 100%

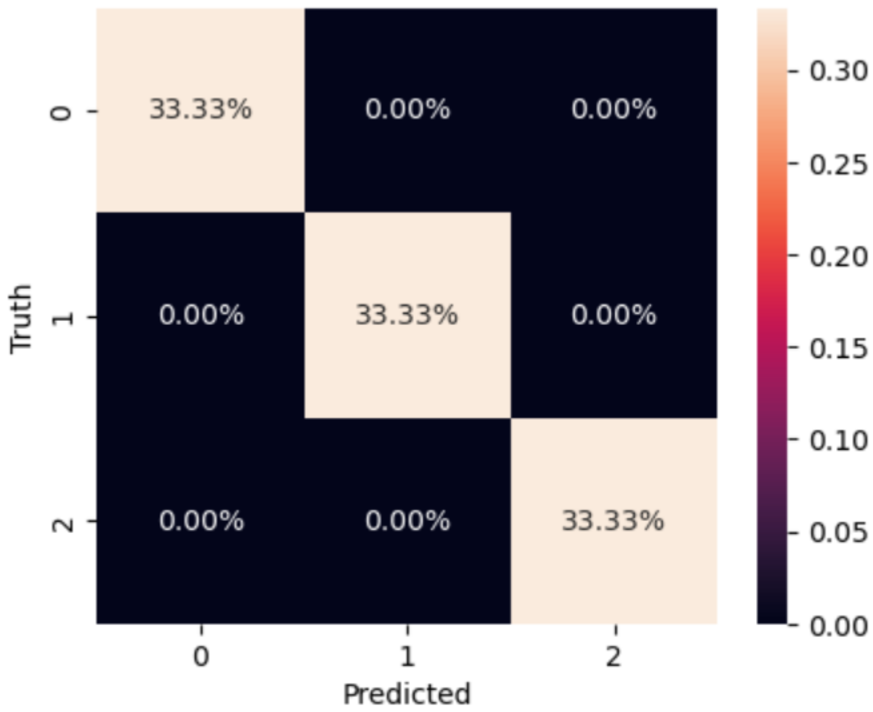
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 7: Confusion Matrix for Bayes Classifier's Case 2 Part 2 implemented on Linearly Separable Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 21, while the decision region plot for all classes along with the testing data superimposed in shown in Figure 22.

As observed from the figures, the results are similar to the previous case. The decision boundaries formed are linear, and all the samples in the testing data are classified accurately, as seen from the confusion matrix in Table 7.

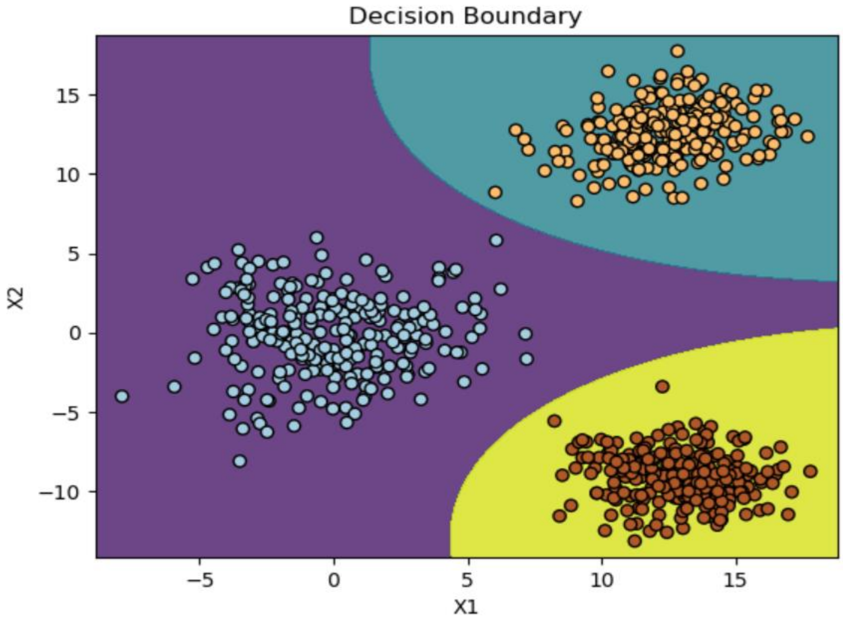


Figure 21: Decision region plot for all classes along with the training data superimposed as obtained by the Bayes Classifier's Case 2 Part 2

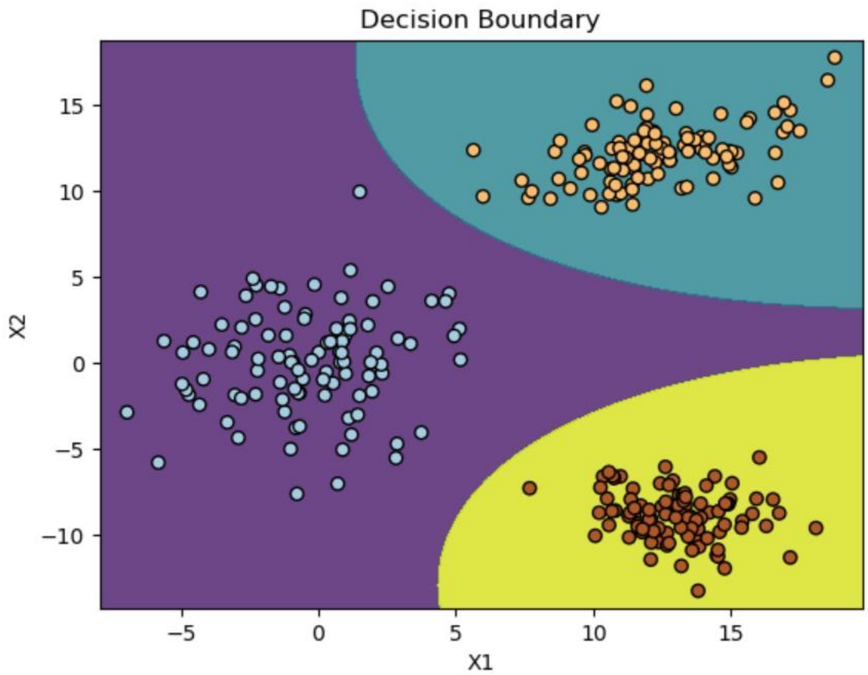


Figure 22: Decision region plot for all classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 2 Part 2

2.4.4 Covariance matrix is diagonal and is different for each class

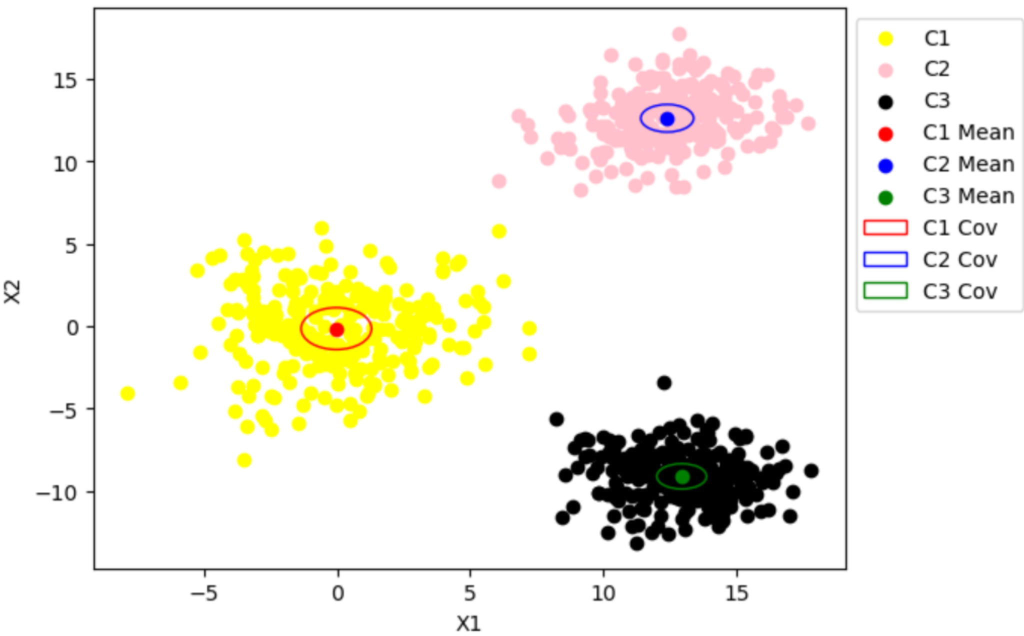


Figure 23: Covariance matrix is diagonal and is different for each class

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the off diagonal elements for each covariance matrix have been made 0. The classifier's performance was measured using the following performance measures.

Their values are listed below:

Accuracy on Testing Set: 100%

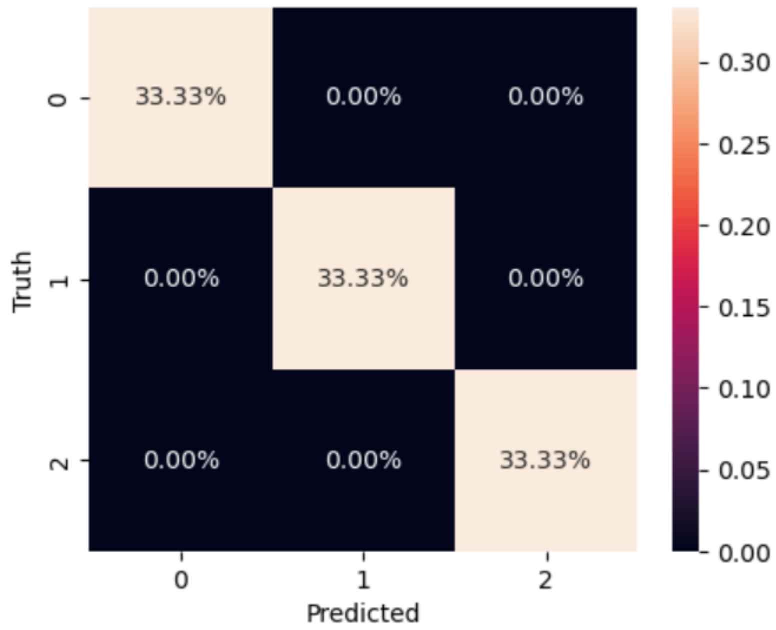
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 8: Confusion Matrix for Bayes Classifier's Case 3 implemented on Linearly Separable Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 24 , while the decision region plot for all classes along with the testing data superimposed in shown in Figure 25.

As observed from the figures below, interestingly, the decision boundaries formed are not linear. However, all the samples in the testing data are classified accurately, as observed from the confusion matrix in Table 8.

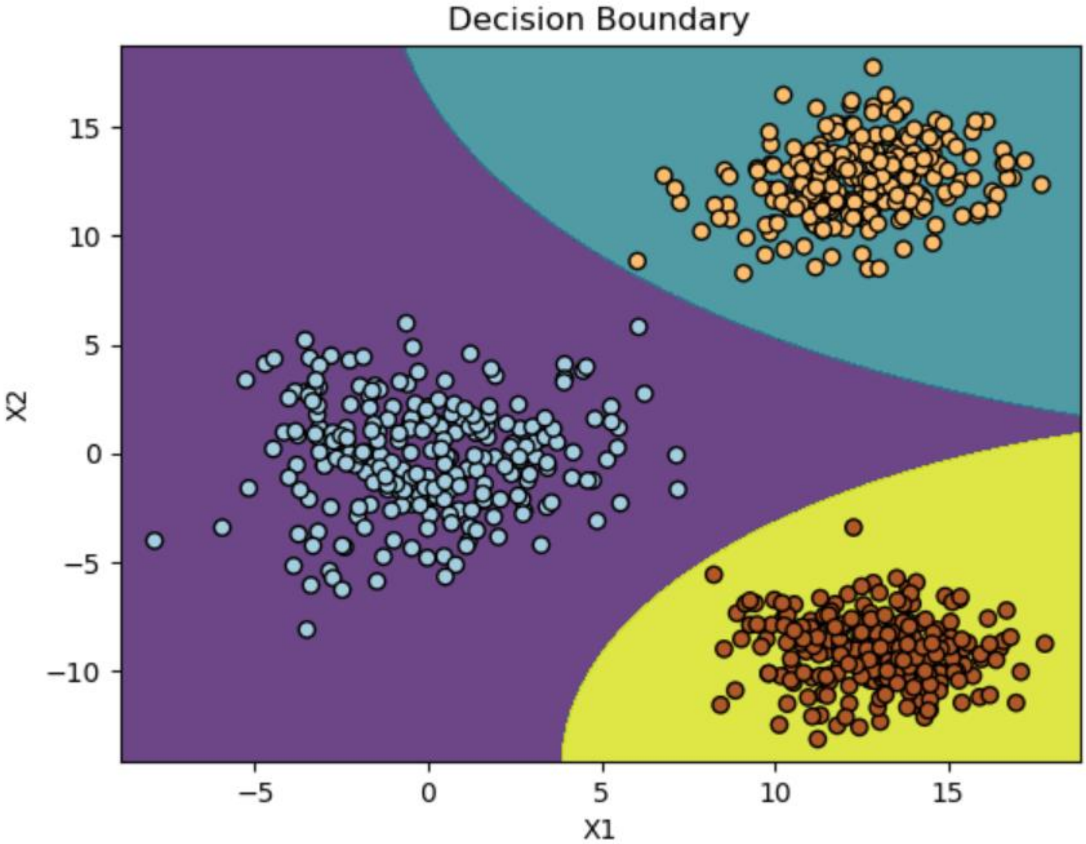


Figure 24: Decision region plot for all classes along with the training data superimposed as obtained by the Bayes Classifier's Case 3

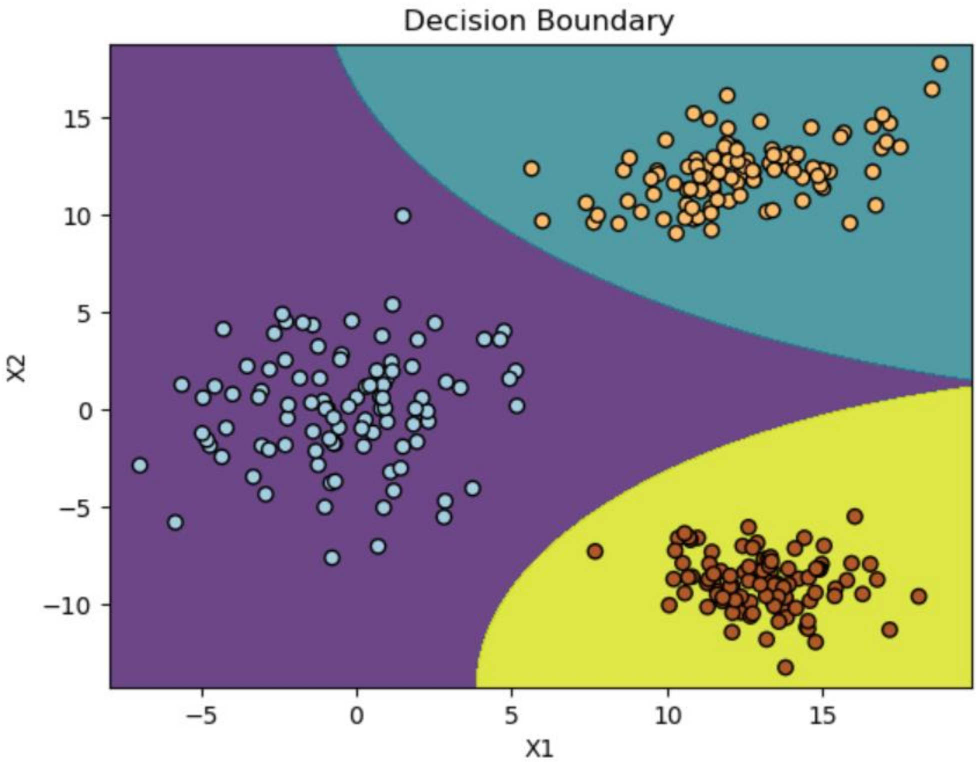


Figure 25: Decision region plot for all classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 3

2.4.5 Full covariance matrix for each class and is different

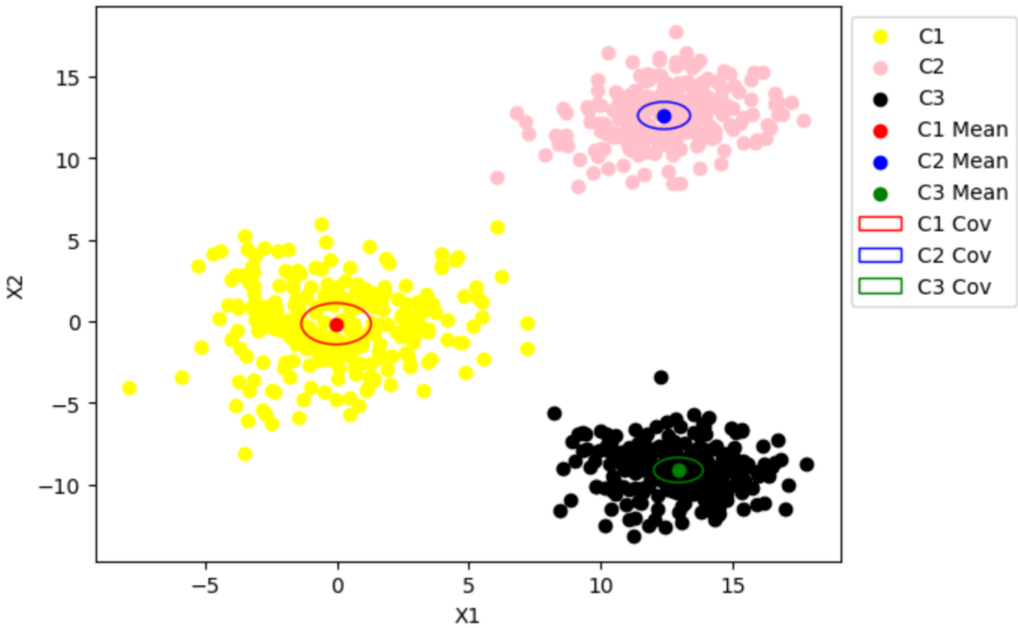


Figure 26: Full covariance matrix for each class and is different

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the full covariance matrix is taken. The classifier's performance was measured using the following performance measures. Their values are listed below:

Accuracy on Testing Set: 100%

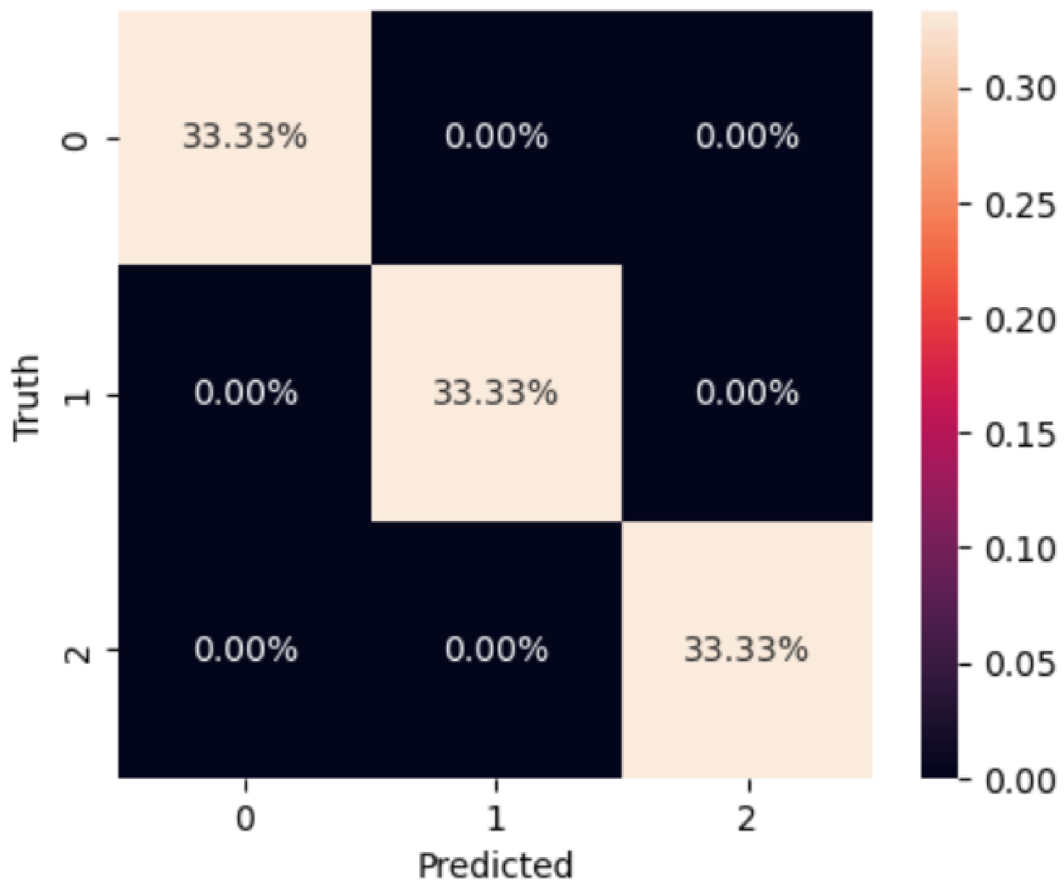
Mean Precision: 100%

Mean Recall: 100%

Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 9: Confusion Matrix for Bayes Classifier's Case 4 implemented on Linearly Separable Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 27, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 28.

As observed from the figures, the decision boundaries formed are not linear. This is because we have taken different covariance matrices for all the classes. However, once again, all the samples in the testing data are classified correctly, as observed from the confusion matrix in Table 9.

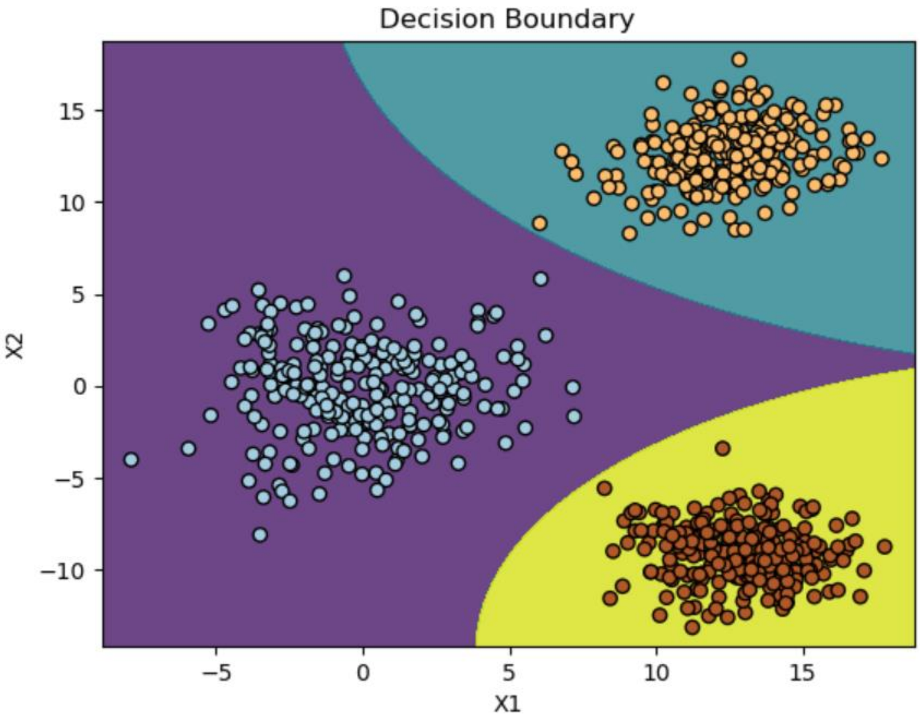


Figure 27: Decision region plot for all classes along with the training data superimposed as obtained by the Bayes Classifier's Case 4

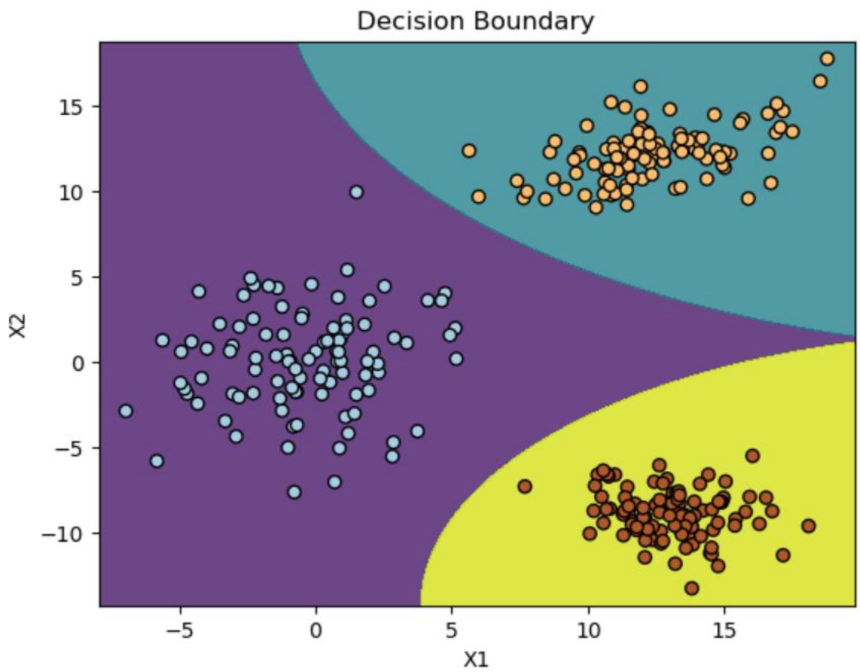


Figure 28: Decision region plot for all classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 4

2 Nonlinearly Separable Classes

Our dataset consists of two nonlinearly separable classes. We have labeled the features as Feature 1 and Feature 2, while the class labels corresponding to Class 0 is 0 and Class 1 is 1. The red color corresponds to Class 0 while the green color corresponds to Class 1. The training dataset along with the mean for each class is shown in Figure 48. As observed from Figure 48, the mean for class 0 is (-0.04, -1.03) and class 1 is (1.08, 2.11). The data shows two classes that are of a spiral nature, but separable by a nonlinear decision boundary. The means for both the classes fall outside the class data.

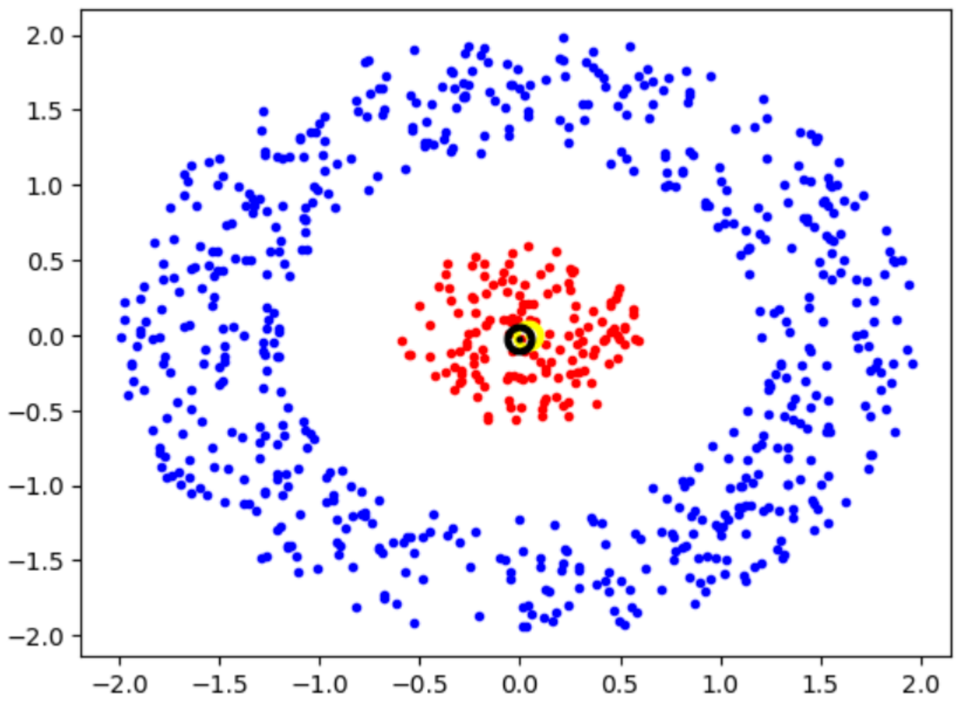


Figure 29: Training data of the nonlinearly separable class dataset along with the means of each class

We now try to apply the Nearest Neighbour classifier to see the decision boundaries and accuracy of the classifier.

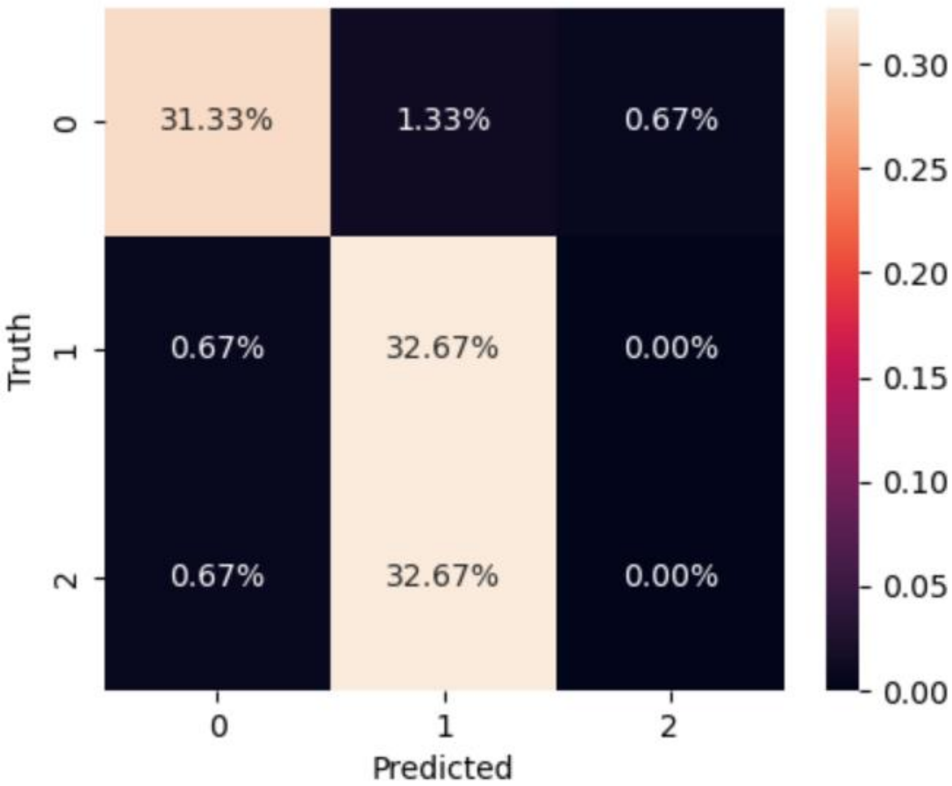
2.1 Nearest Neighbour Classifier

We have implemented the Nearest Neighbour Classifier in Python using the KNeigboursClassifier model from sklearn. To model the nearest neighbour classifier, we have set the value of K to be 1 for this case. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Test Set: 100%
Accuracy on Validation Set: 100%
Mean Precision: 100%
Mean Recall: 100%
Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 10: Confusion Matrix for Nearest Neighbour Classifier implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 30, while the decision region plot for both the classes along with the testing data superimposed in shown in Figure 31.

As observed from figures 30 and 31, the decision boundaries formed are nonlinear, and the classifier gives a 100% accuracy as there is no sample that is classified incorrectly.

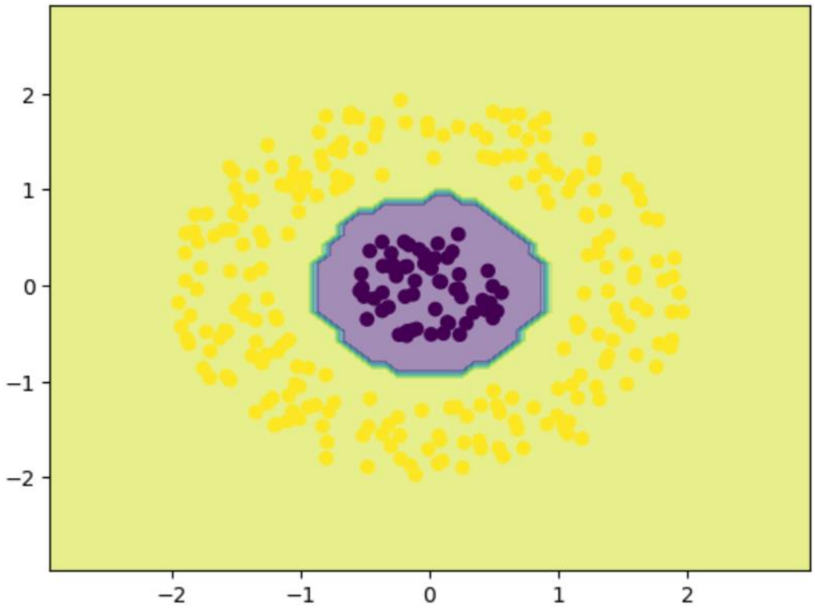


Figure 30: Decision region plot for both the classes superimposed with the training data

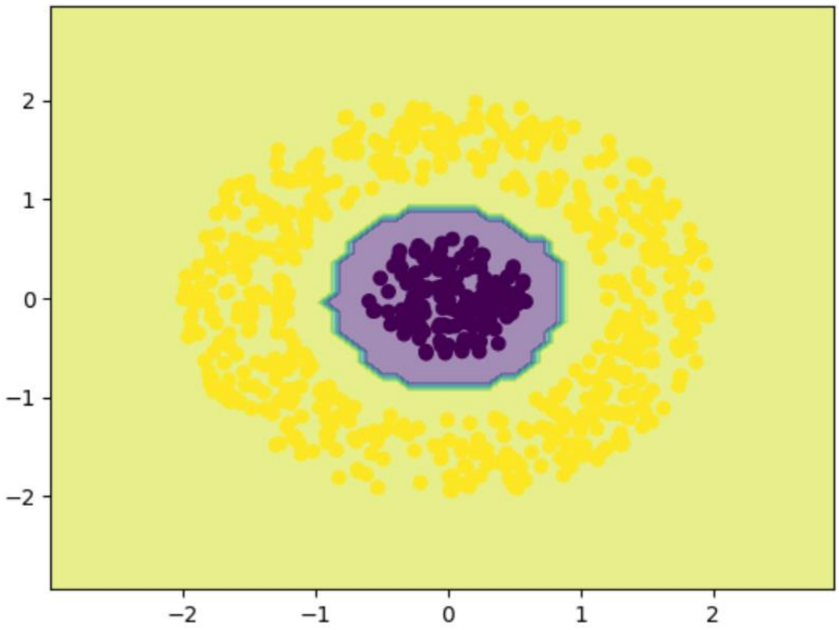


Figure 31: Decision region plot for both the classes superimposed with the testing data

2.2 K-Nearest Neighbour Classifier

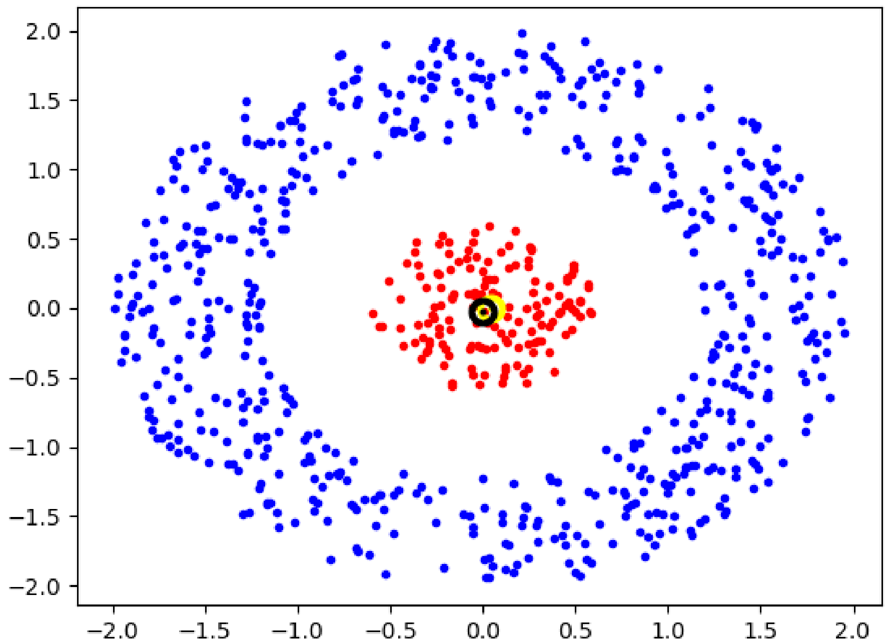


Figure 32: K-Nearest Neighbour Classifier

We have implemented the K- Nearest Neighbour Classifier in Python using the KNeigboursClassifier model from sklearn. In order to find the K value that is optimal for our case, we have fit the classifier for increasing values of K up to half of the length of the training dataset. The plot of K values vs. accuracy is shown below:

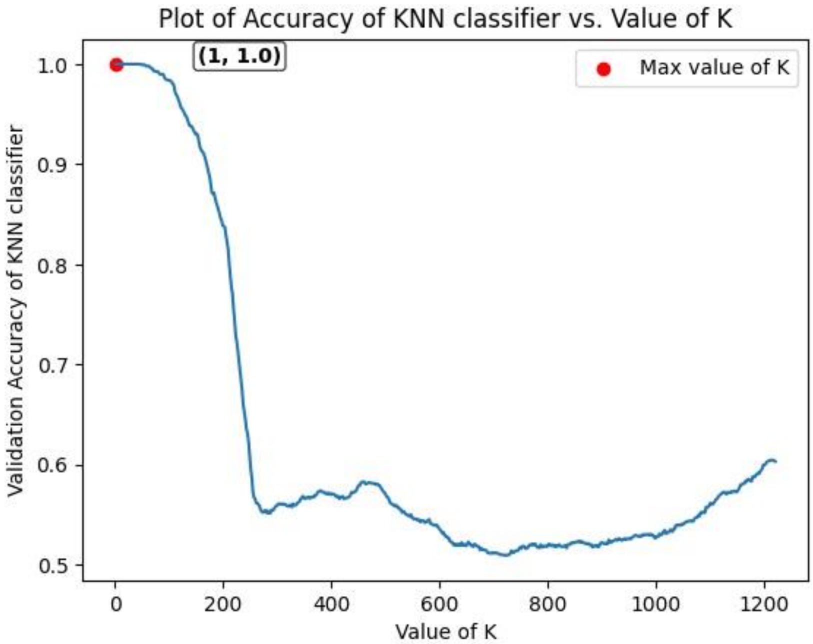


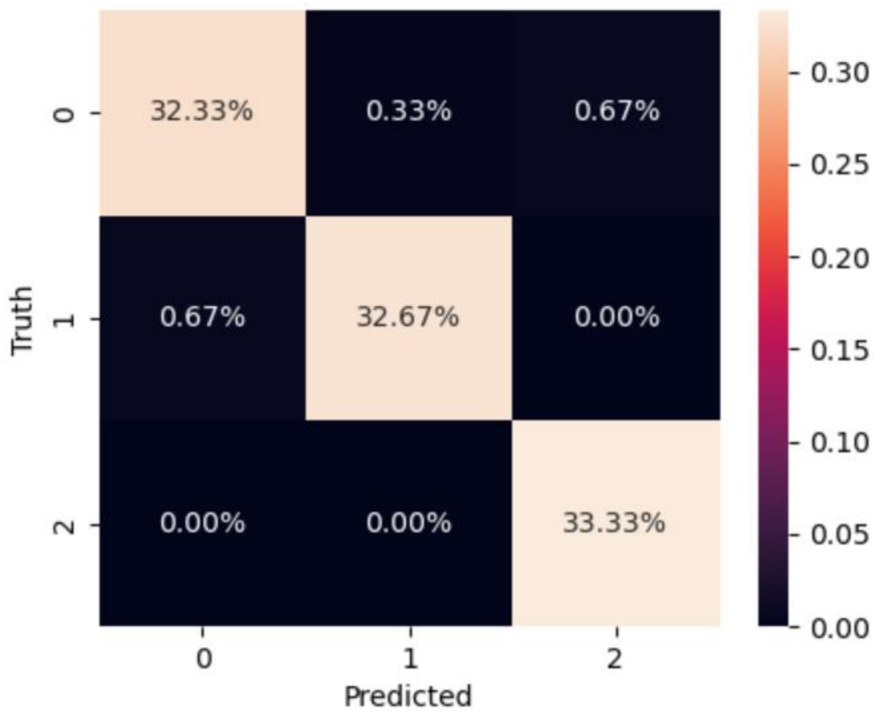
Figure 33: Plot of Accuracy of the KNN classifier on the Validation Set vs. the value of K

As observed from Figure 33, the accuracy is maximum (100%) at K=1. We have therefore used K=1 to train our classifier. This would basically result in the same implementation as the Nearest Neighbour Classifier. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Test Set: 100%
Accuracy on Validation Set: 100%
Mean Precision: 100%
Mean Recall: 100%
Mean F1-score on Testing Dataset: 100%

Confusion Matrix:

Table 11: Confusion Matrix for K-Nearest Neighbour Classifier implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 34, while the decision region plot for both the classes along with the testing data superimposed in shown in Figure 35. The decision boundaries for both the NN and KNN classifier are same in this case.

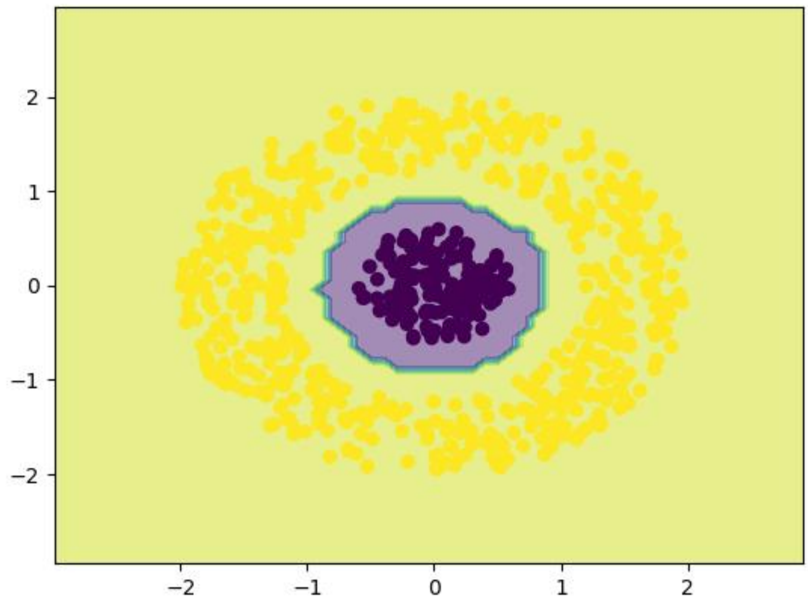


Figure 34: Decision region plot for both the classes superimposed with the training data

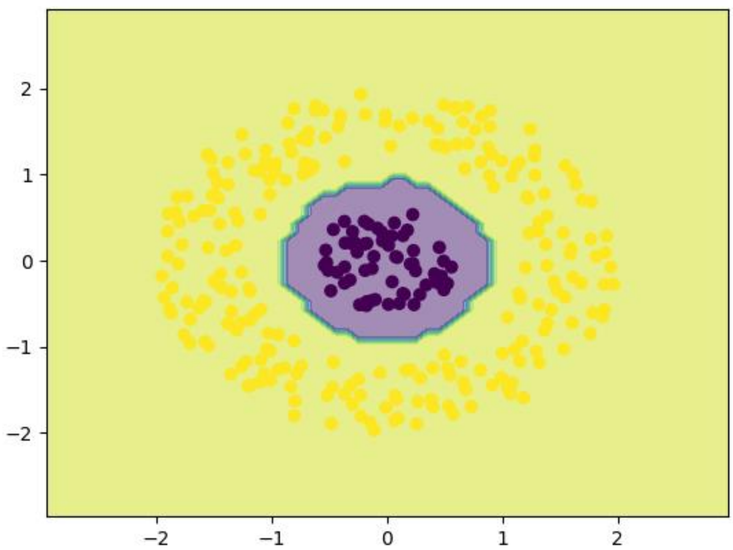


Figure 35: Decision region plot for both the classes superimposed with the testing data

2.3 Reference Template-Based Classifier

We have implemented the reference template-based classifier for both the sample mean and sample mean and covariance matrix-based classifier.

3.3.1 Mean Vector as Reference Template for a Class

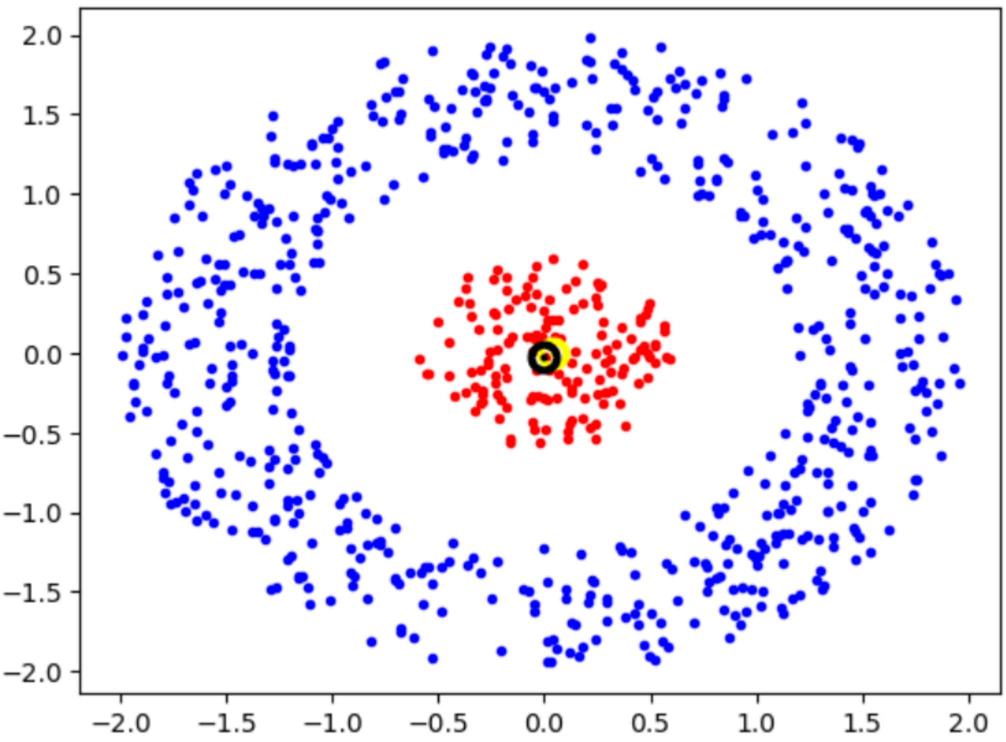


Figure 36: Mean Vector as Reference Template for a Class

We have implemented our own function for the mean vector-based classifier. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Training Set: 62.55%

Accuracy on Testing Set: 63.52%

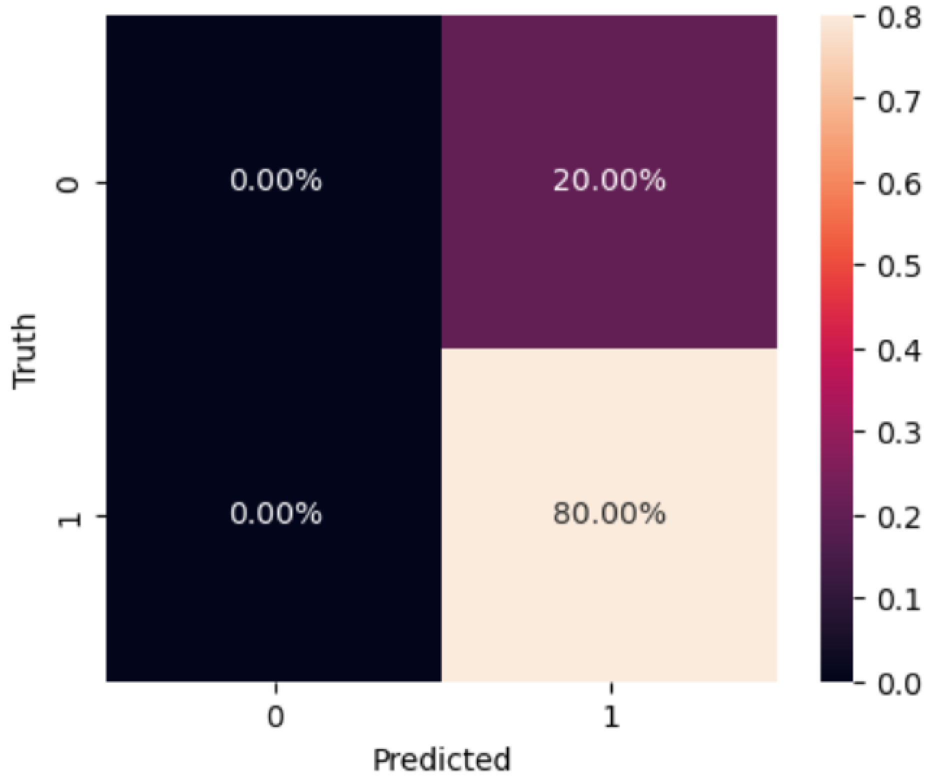
Mean Precision: 63.52%

Mean Recall: 63.52%

Mean F1-score on Testing Dataset: 63.52%

Confusion Matrix:

Table 12: Confusion Matrix for Mean Vector-based Classifier implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 37, while the decision region plot for both the classes along with the testing data superimposed in shown in Figure 38.

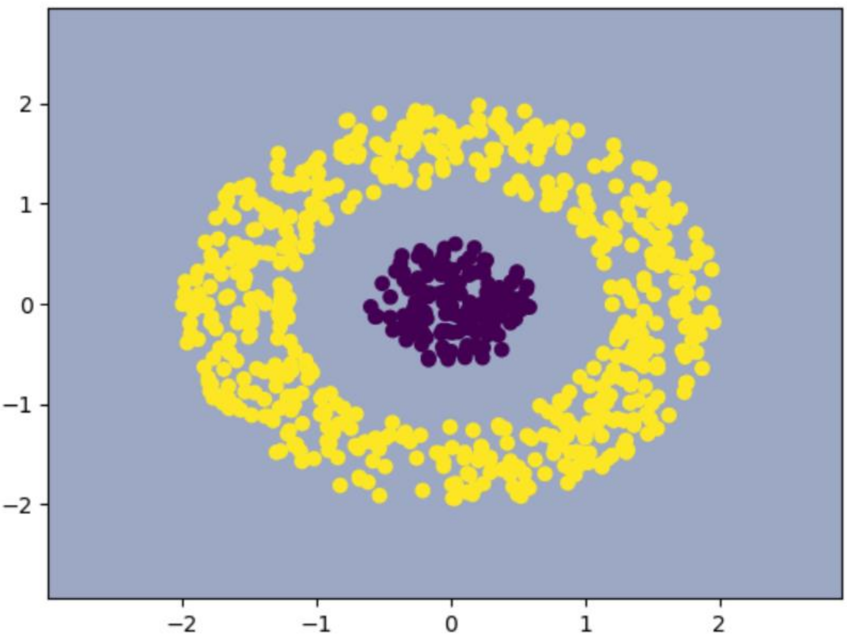


Figure 37: Decision region plot for both the classes superimposed with the training data for Mean Vector-based classifier

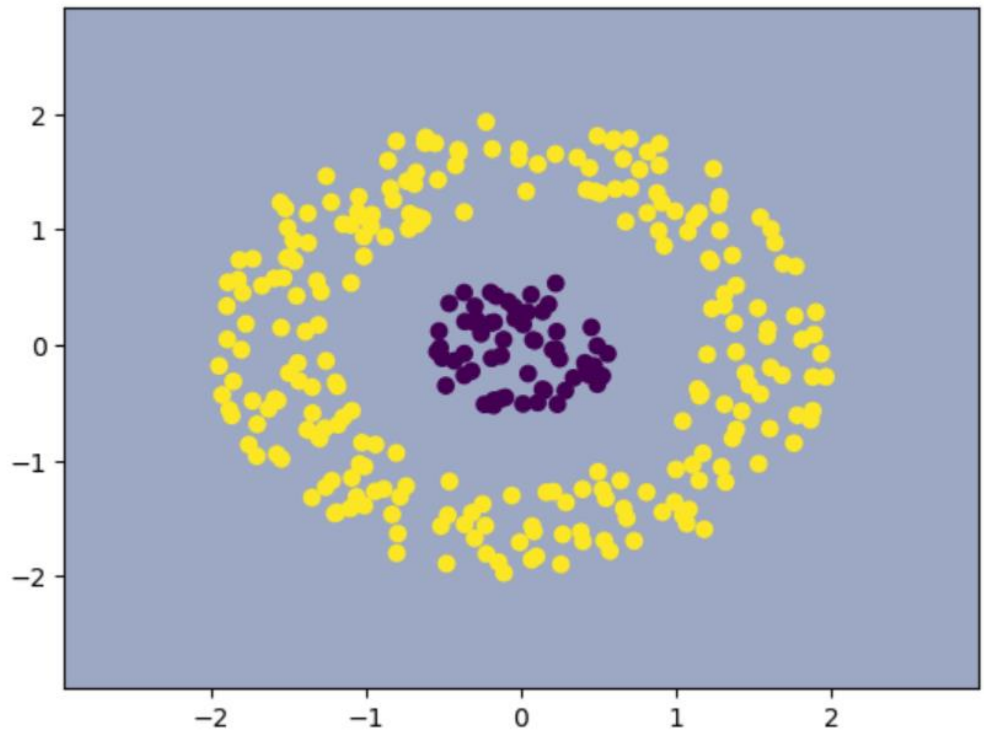


Figure 38: Decision region plot for both the classes superimposed with the testing data for Mean Vector-based classifier

As observed from figures 37 and 38, the mean vector-based classifier forms a linear decision boundary, which is clearly not a good fit for our nonlinearly separable classes. This explains the low values of our performance measures on this classifier.

3.3.2 Mean Vector and Covariance Matrix as Reference Template for a Class

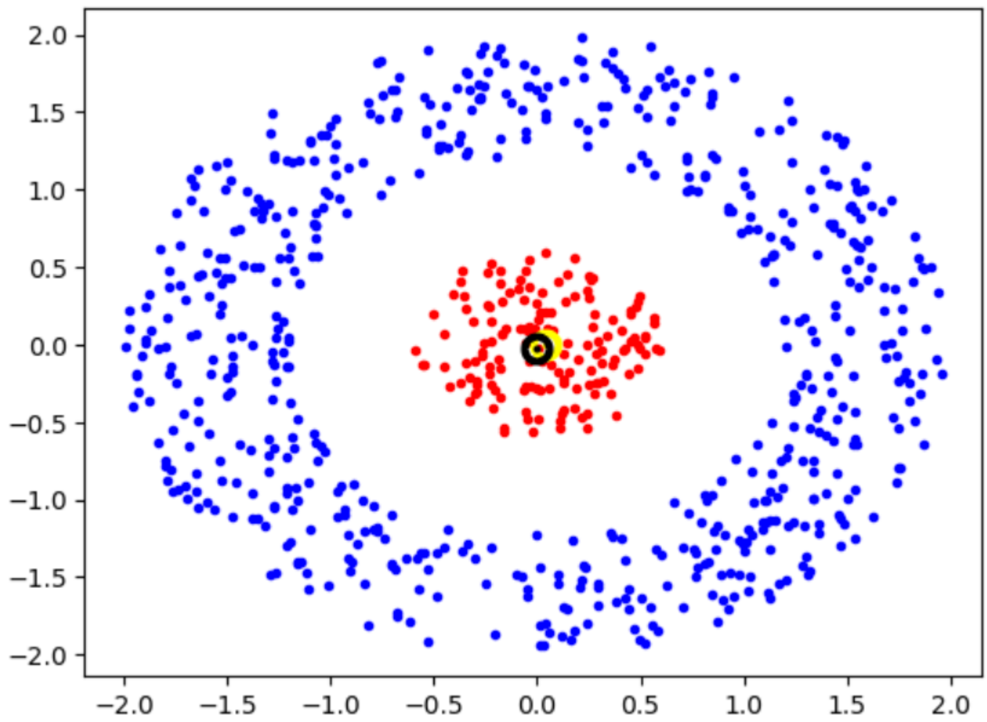


Figure 39: Mean Vector and Covariance Matrix as Reference Template for a Class

We have implemented our own function for the mean vector and covariance matrix-based classifier. The Mahalonobis distance metric was imported from the SciPy library. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 63.25%

Accuracy on Validation Set: 62.28%

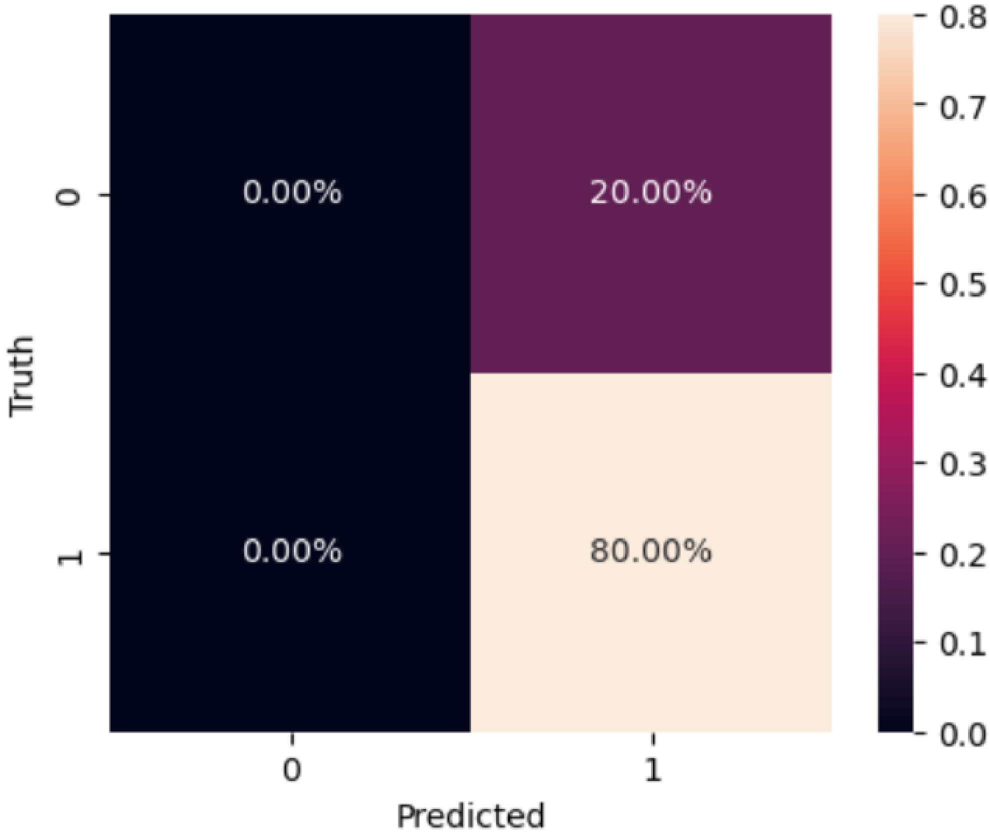
Mean Precision: 63.27%

Mean Recall: 63.25%

Mean F1-score on Testing Dataset: 63.24%

Confusion Matrix:

Table 13: Confusion Matrix for Mean Vector and Covariance Matrix-based Classifier implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 40, while the decision region plot for both the classes along with the testing data superimposed in shown in Figure 41. The results are similar to the Mean Vector-based classifier. The Mean Vector and Covariance Matrix-based classifier also forms a linear decision boundary for the nonlinearly separable classes, thus resulting in poor performance.

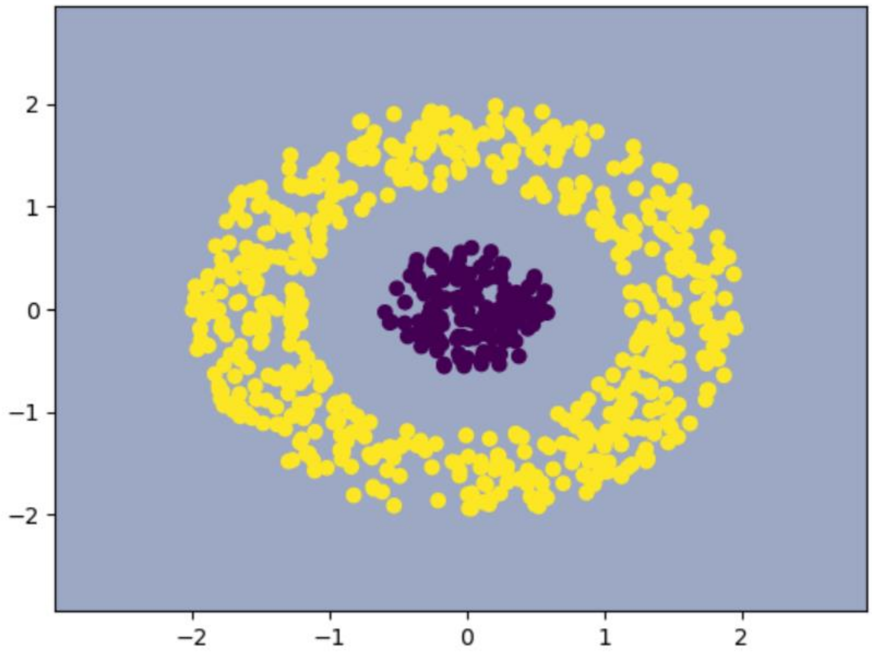


Figure 40: Decision region plot for both the classes superimposed with the training data for Mean Vector and Covariance Matrix-based classifier

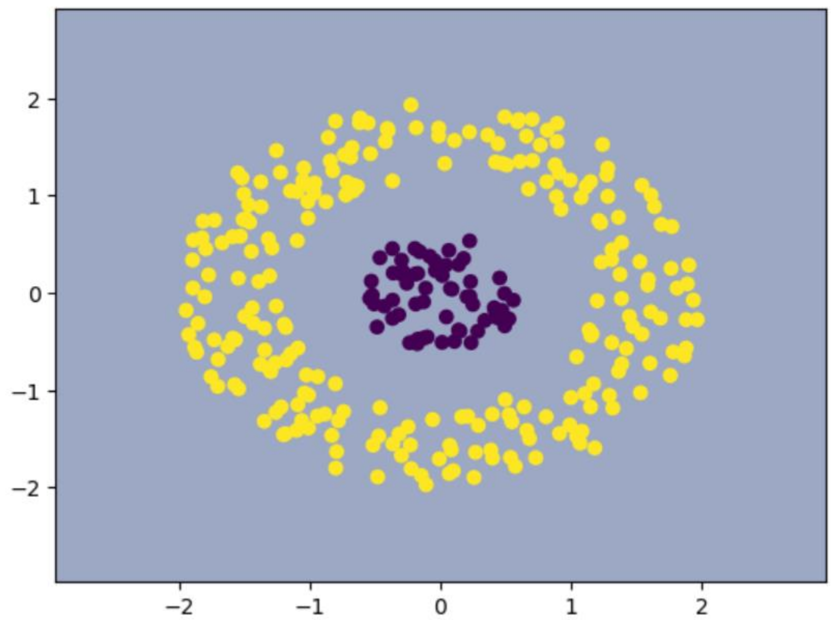


Figure 41: Decision region plot for both the classes superimposed with the testing data for Mean Vector and Covariance Matrix-based classifier

2.4 Bayes Classifier-Unimodal Gaussian Density

Bayes Classifier was implemented for all four cases of the covariance matrix. The four cases are listed below.

3.4.1 Covariance matrix for all the classes is the same and is $\sigma^2 I$

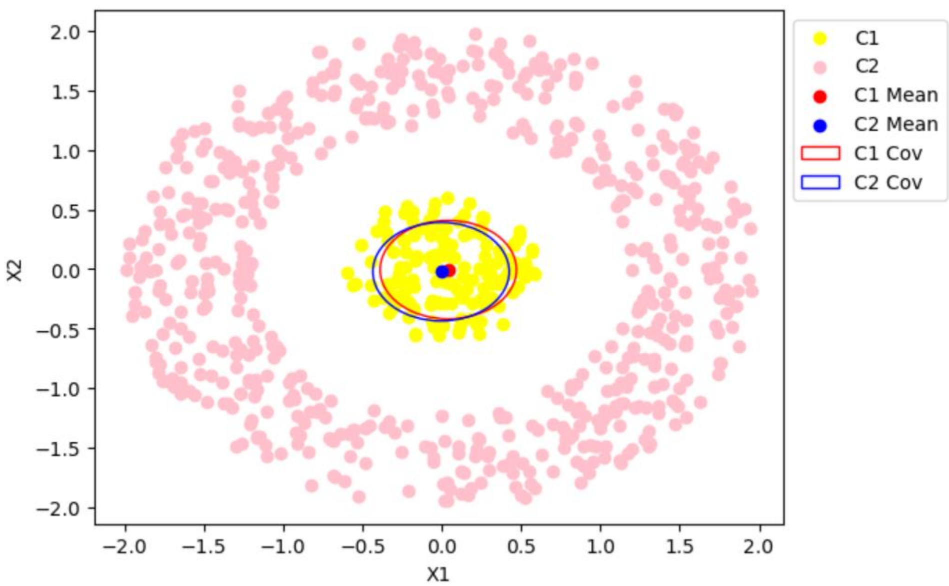


Figure 42: Covariance matrix for all the classes is the same and is σ2I

The features are independent in this case and both the features have the same variance. The Gaussian density function was implemented using the *multivariate_normal* module from SciPy.stats. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 63.114%

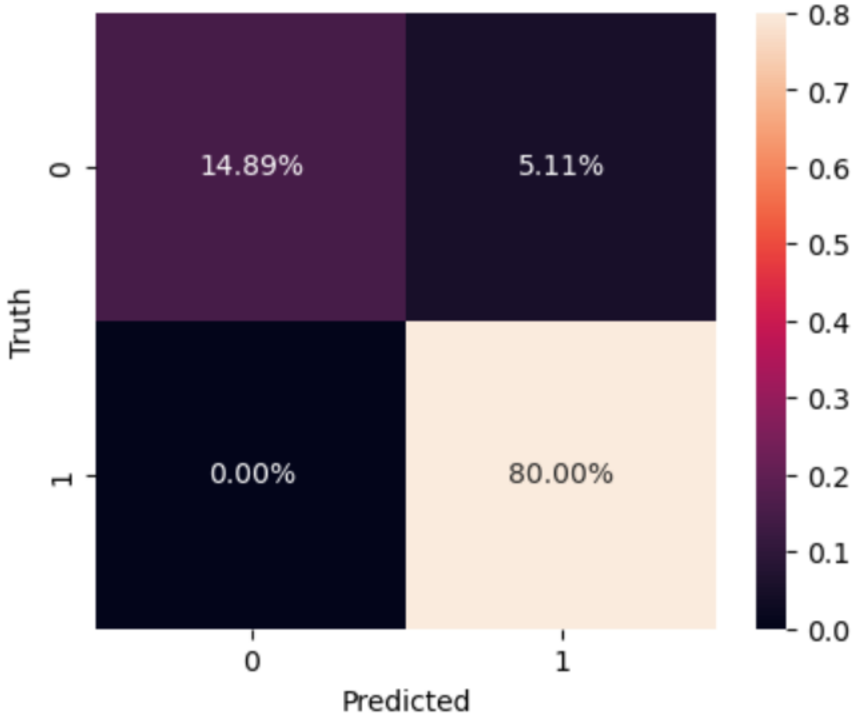
Mean Precision: 63.12%

Mean Recall: 63.114%

Mean F1-score on Testing Dataset: 63.109%

Confusion Matrix: The decision region plot for both the classes along with

Table 14: Confusion Matrix for Bayes Classifier's Case 1 implemented on Nonlinearly Separable Classes



The training data superimposed is shown in Figure 43, while the decision region plot for both the classes along with the testing data superimposed is shown in Figure 44.

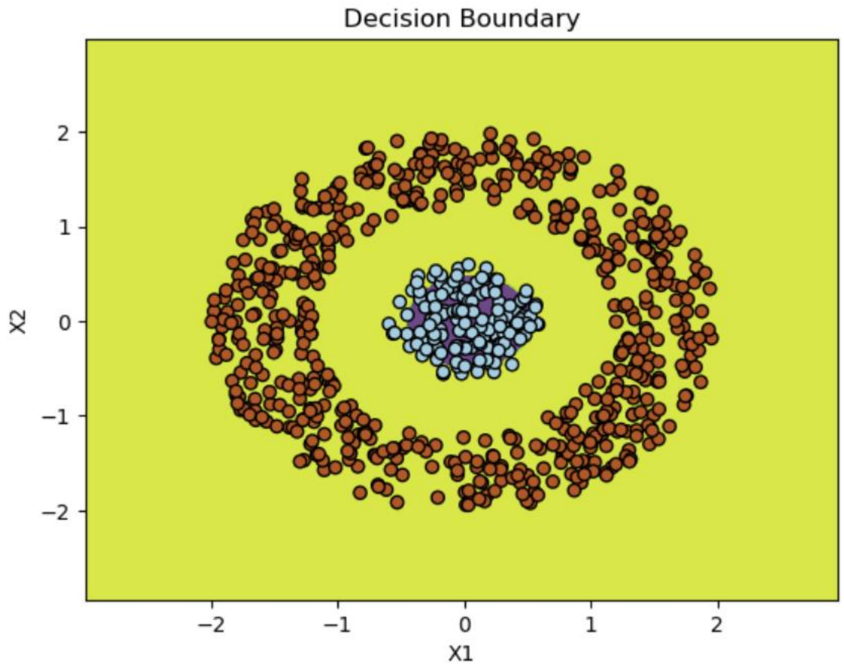


Figure 43: Decision region plot for both the classes superimposed with the training data for Bayes Classifier’s Case 1

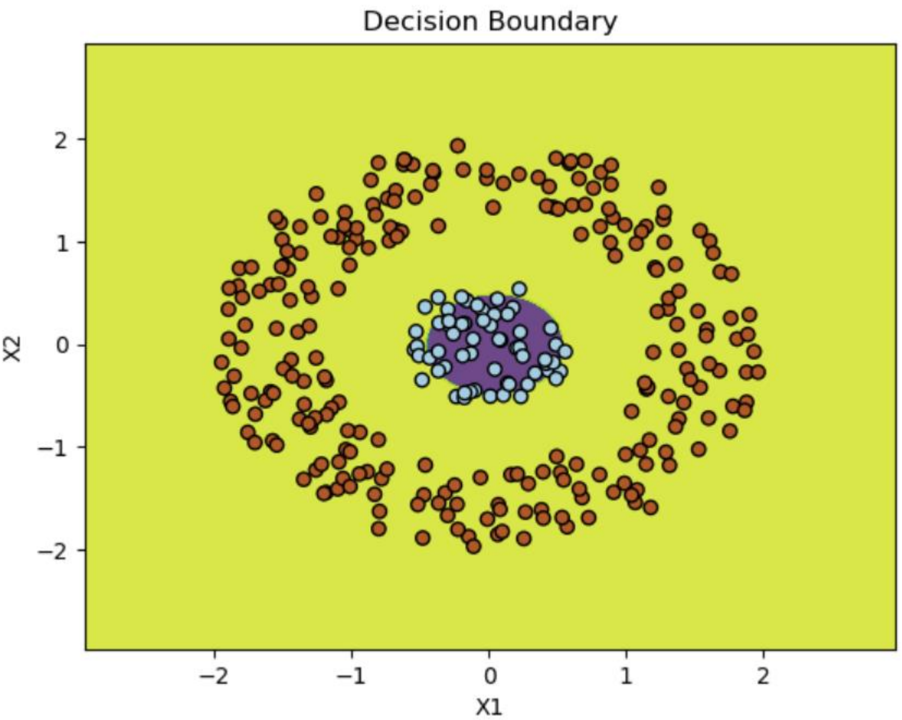


Figure 44: Decision region plot for both the classes superimposed with the testing data for Bayes Classifier's Case 1

As observed from figures 43 and 44, the results are quite similar to the reference template-based classifiers. The Bayesian Classifier also gives a linear decision boundary for Case 1, resulting in poor performance.

3.4.2 Full covariance matrix for all the classes and is same for all the classes - Part 1

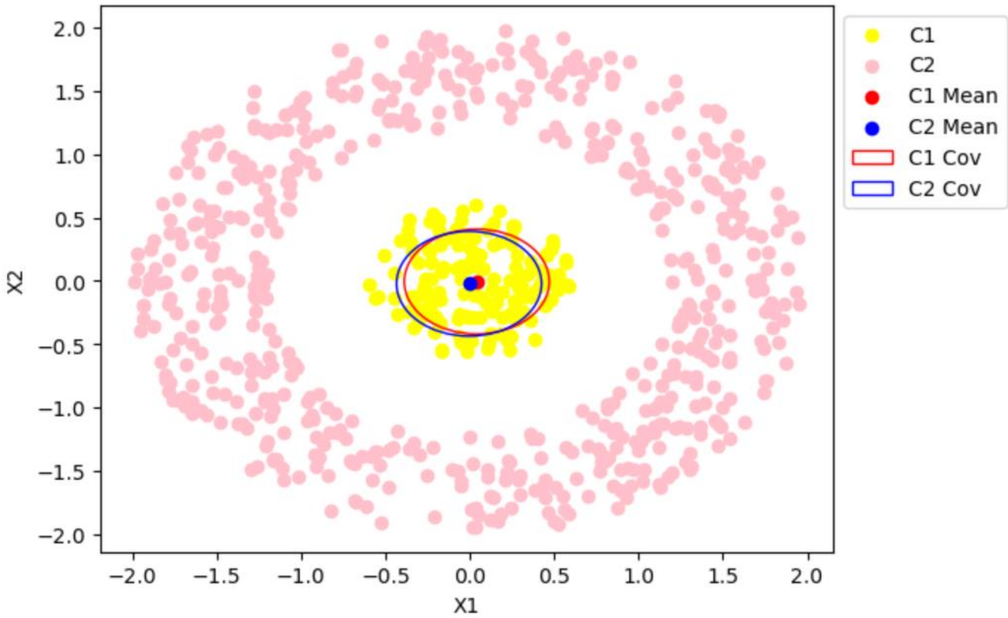


Figure 45: Full covariance matrix for all the classes and is same for all the classes - Part 1

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by taking the average of covariance matrices of all the classes. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 63.114%

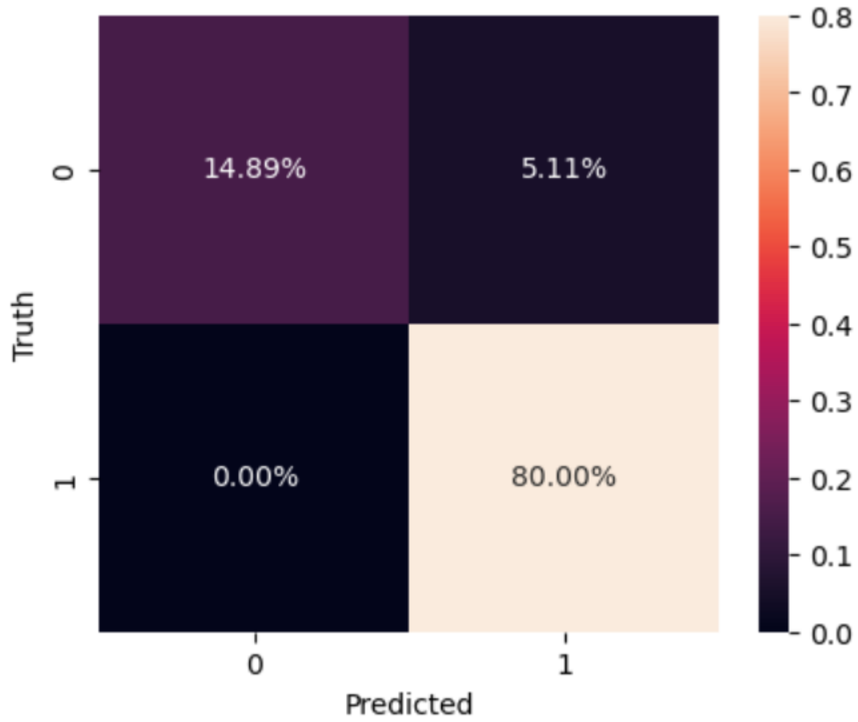
Mean Precision: 63.125%

Mean Recall: 63.114%

Mean F1-score on Testing Dataset: 63.10%

Confusion Matrix:

Table 15: Confusion Matrix for Bayes Classifier's Case 2 Part 1 implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 46, while the decision region plot for both the classes along with the testing data superimposed is shown in Figure 47.

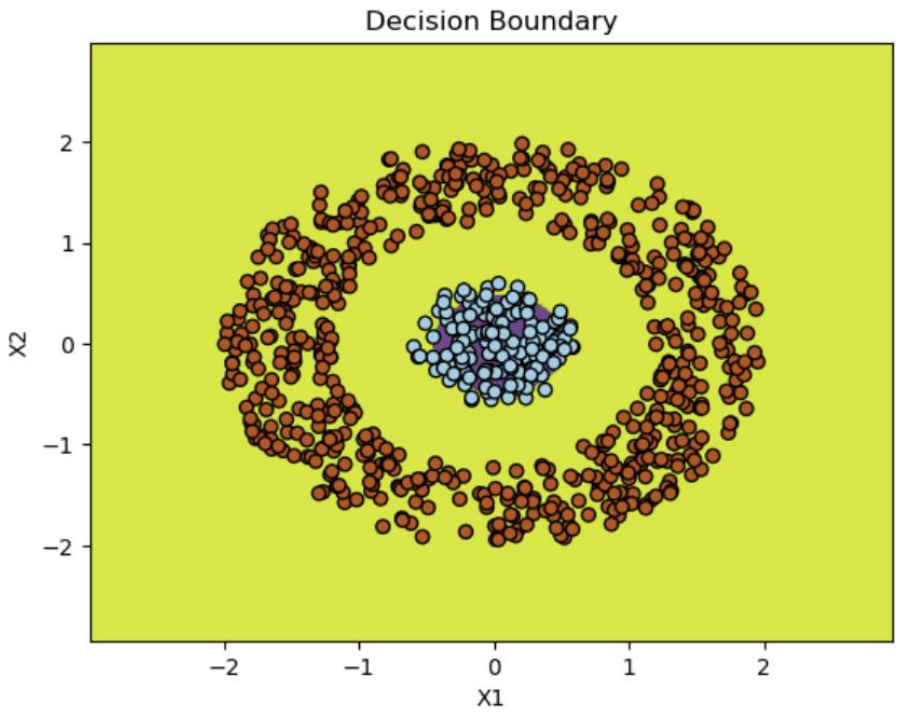


Figure 46: Decision region plot for both the classes superimposed with the training data for Bayes Classifier's Case 2 Part 1

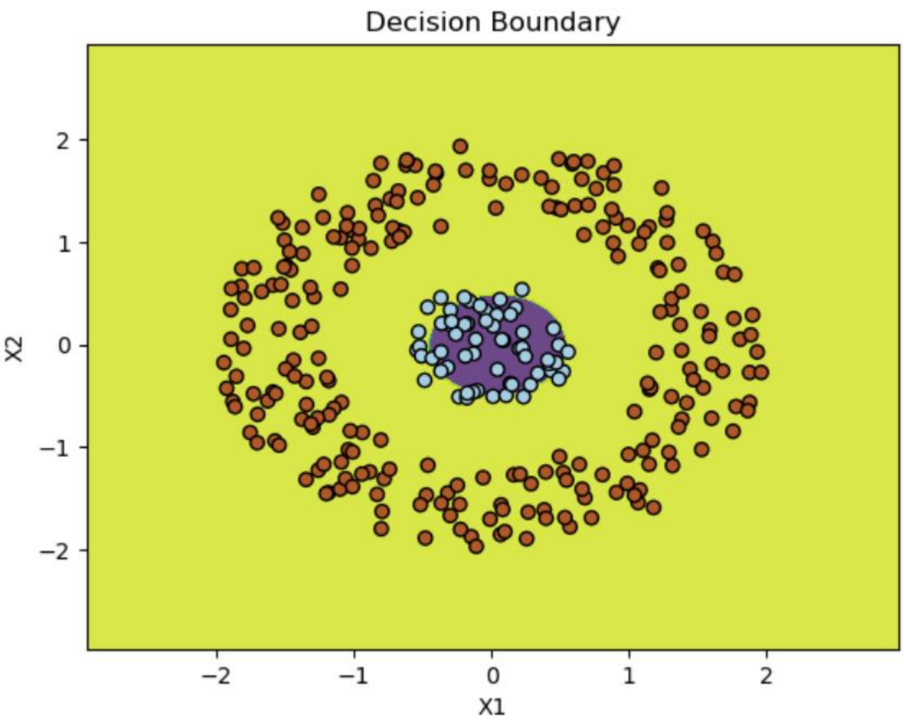


Figure 47: Decision region plot for both the classes superimposed with the testing data for Bayes Classifier's Case 2 Part 1

As observed from figures 46 and 47, once again, the classifier forms a linear decision boundary, resulting in a poor performance.

3.4.3 Full covariance matrix for all the classes and is same for all the classes - Part 2

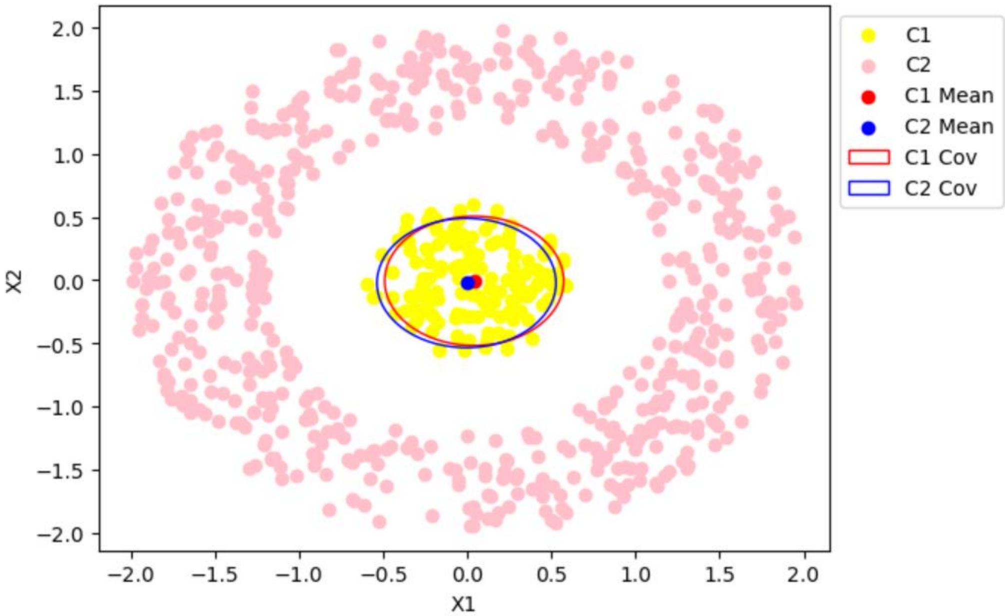


Figure 48: Full covariance matrix for all the classes and is same for all the classes - Part 2

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by computing the covariance matrix of training data of all the classes combined. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 63.114%

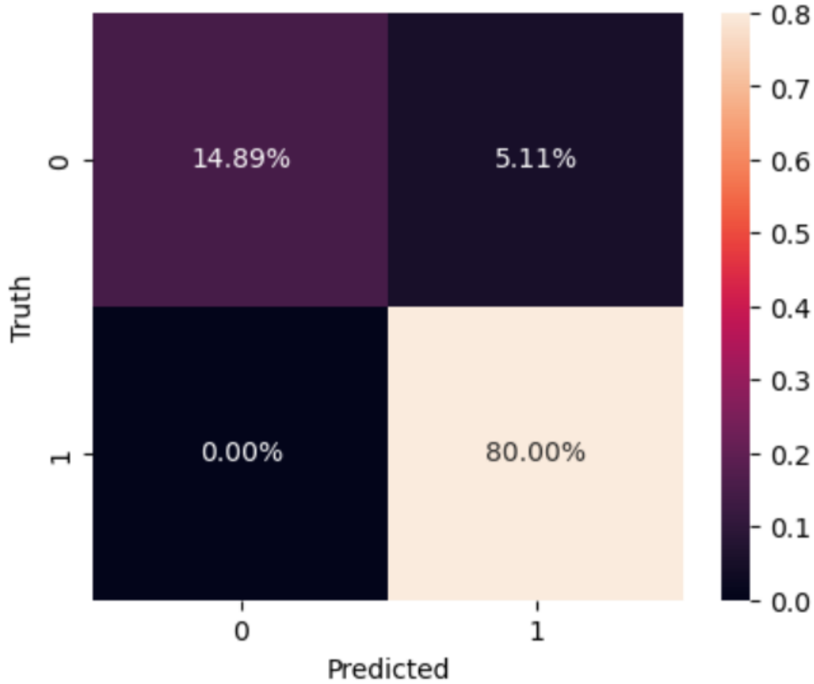
Mean Precision: 63.125%

Mean Recall: 63.114%

Mean F1-score on Testing Dataset: 63.107%

Confusion Matrix:

Table 16: Confusion Matrix for Bayes Classifier's Case 2 Part 2 implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 49, while the decision region plot for both the classes along with the testing data superimposed is shown in Figure 50.

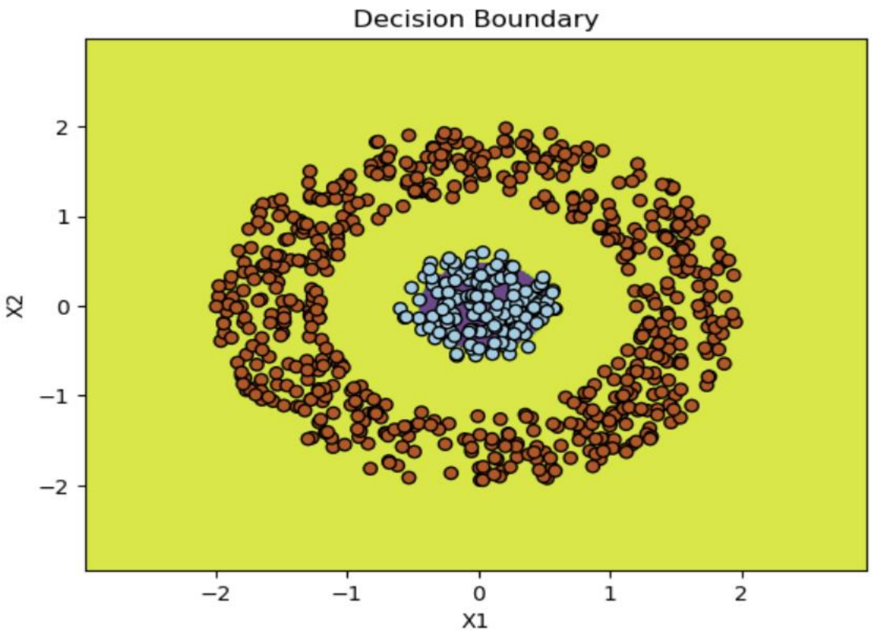


Figure 49: Decision region plot for both the classes superimposed with the training data for Bayes Classifier's Case 2 Part 2

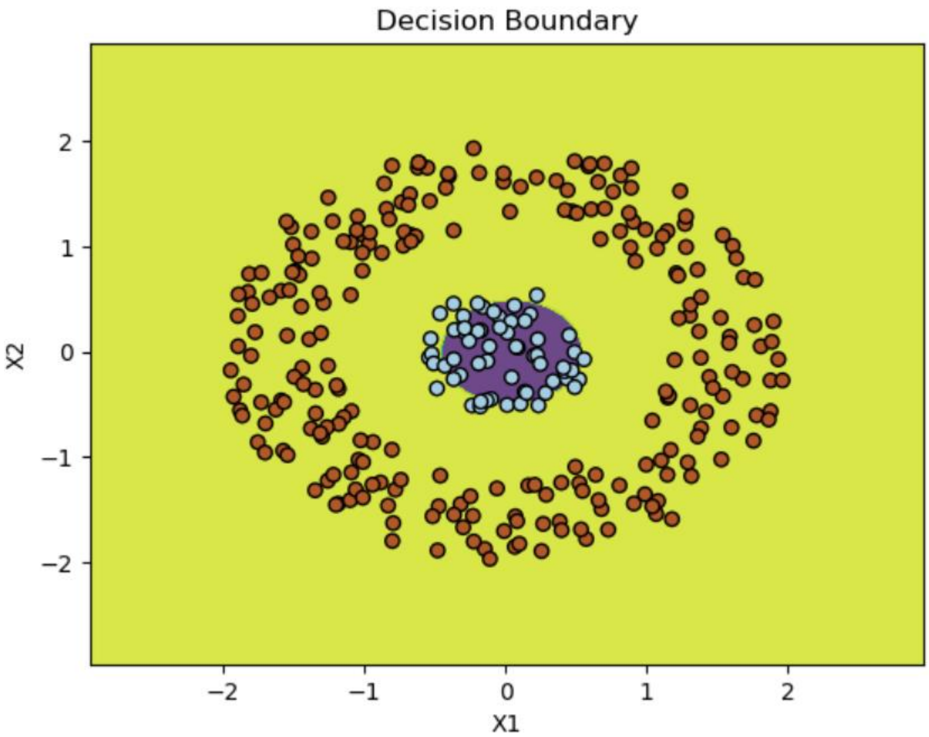


Figure 50: Decision region plot for both the classes superimposed with the testing data for Bayes Classifier's Case 2 Part 2

As observed from figures 49 and 50, the results are similar to Case 2 Part 1. The decision boundary formed is linear, and therefore results in poor performance.

3.4.4 Covariance matrix is diagonal and is different for each class

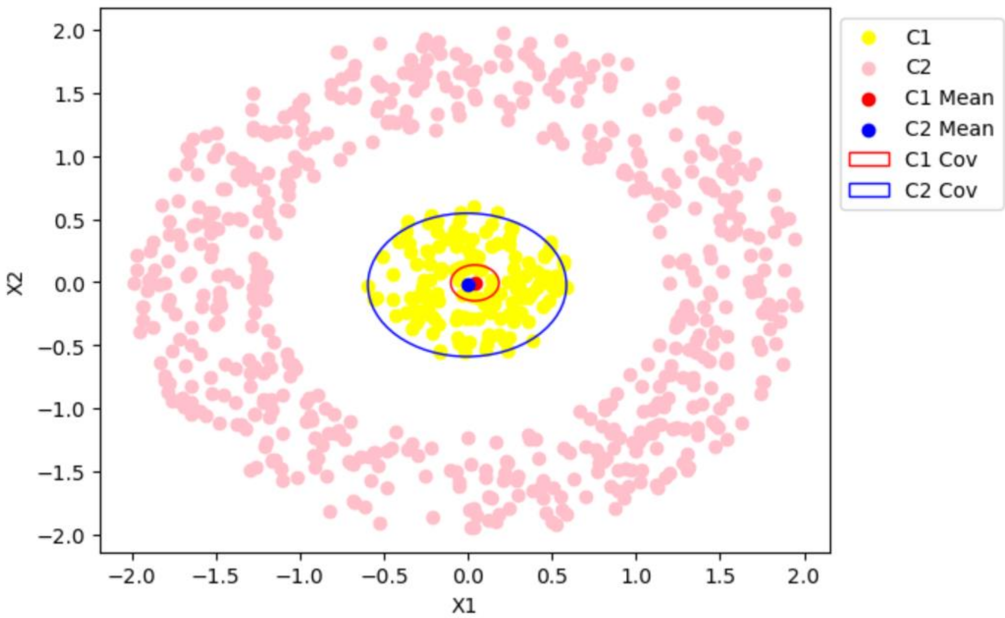


Figure 51: Covariance matrix is diagonal and is different for each class

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the offdiagonal elements for each covariance matrix have been made 0. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 63.42%

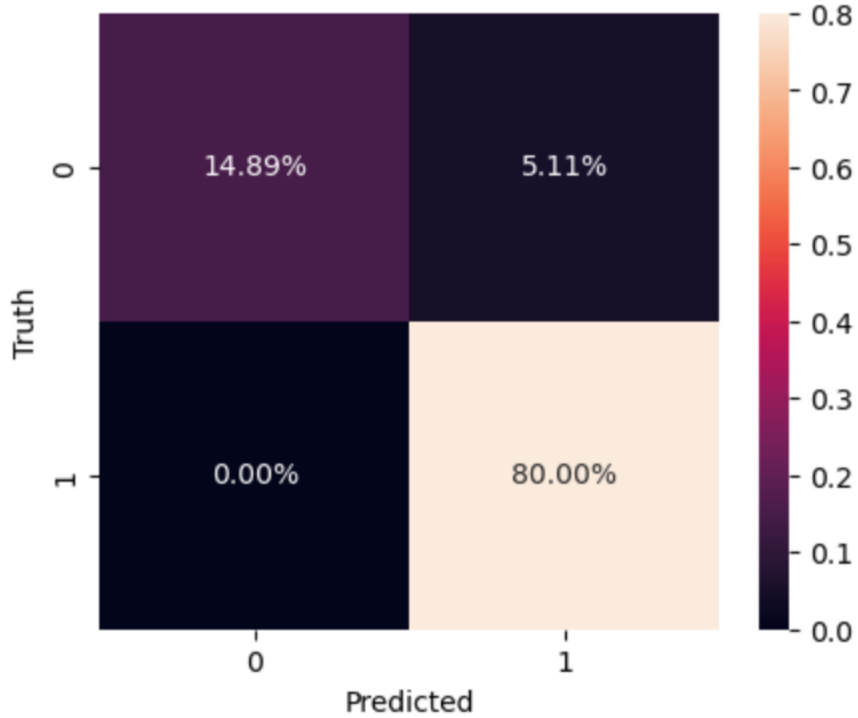
Mean Precision: 63.44%

Mean Recall: 63.42%

Mean F1-score on Testing Dataset: 63.40%

Confusion Matrix:

Table 17: Confusion Matrix for Bayes Classifier's Case 3 implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 52, while the decision region plot for both the classes along with the testing data superimposed is shown in Figure 53.

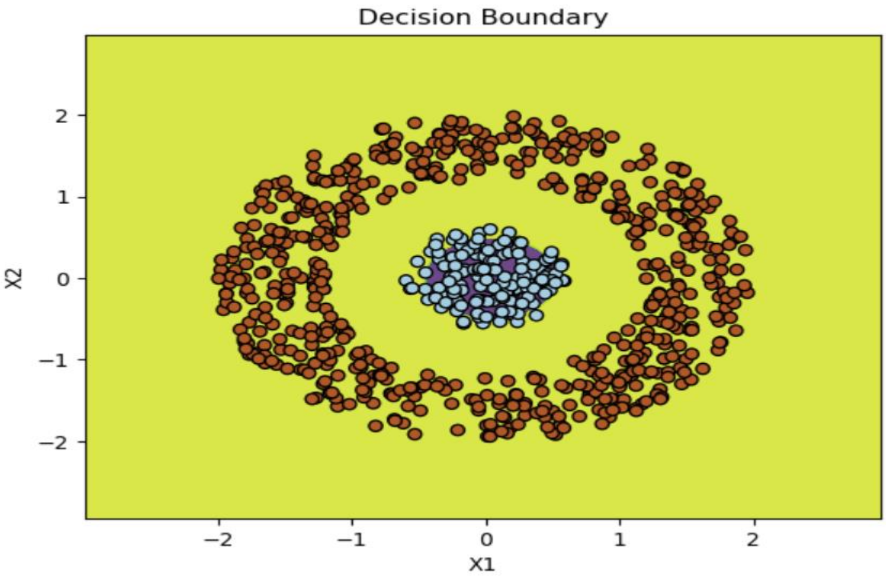


Figure 52: Decision region plot for both the classes superimposed with the training data for Bayes Classifier's Case 3

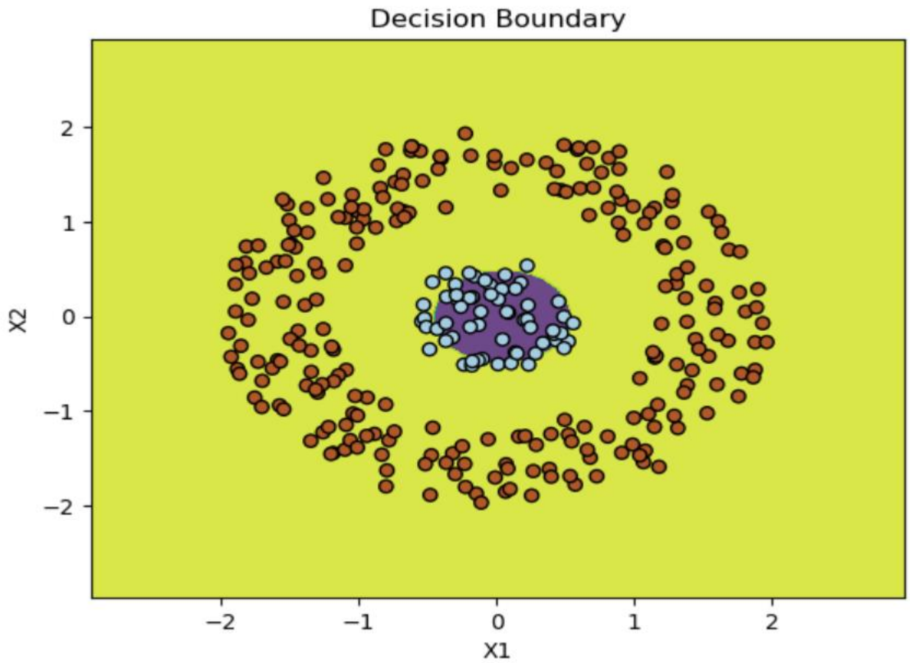


Figure 53: Decision region plot for both the classes superimposed with the testing data for Bayes Classifier's Case 3

As observed from figures 52 and 53, interestingly, the decision boundary formed is not quite linear. This is due to the fcat the covariance matrices for both the classes are taken to be differenr from each other.

3.4.5 Full covariance matrix for each class and is different

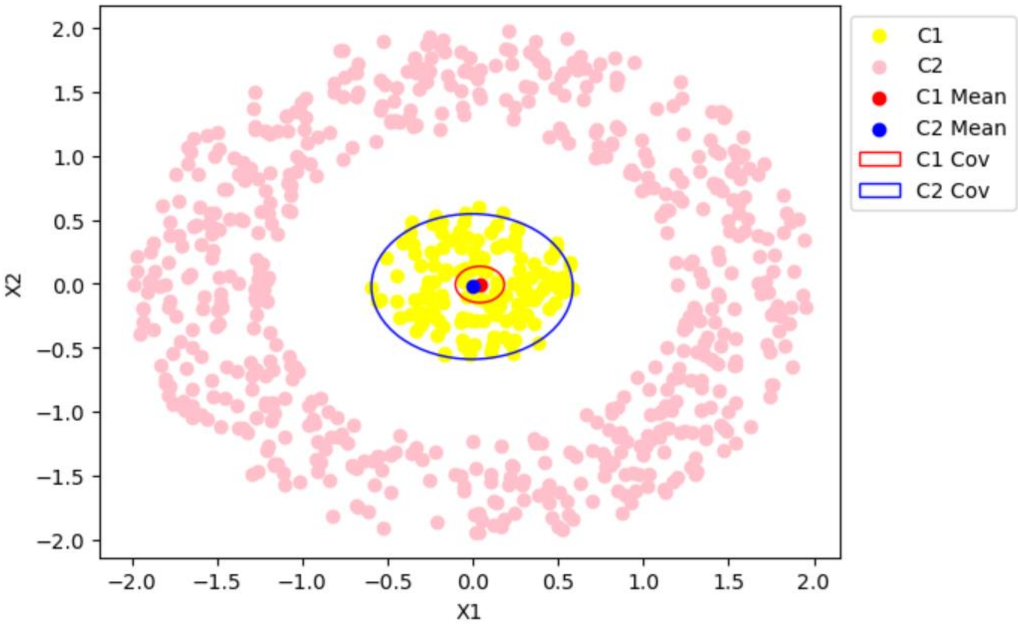


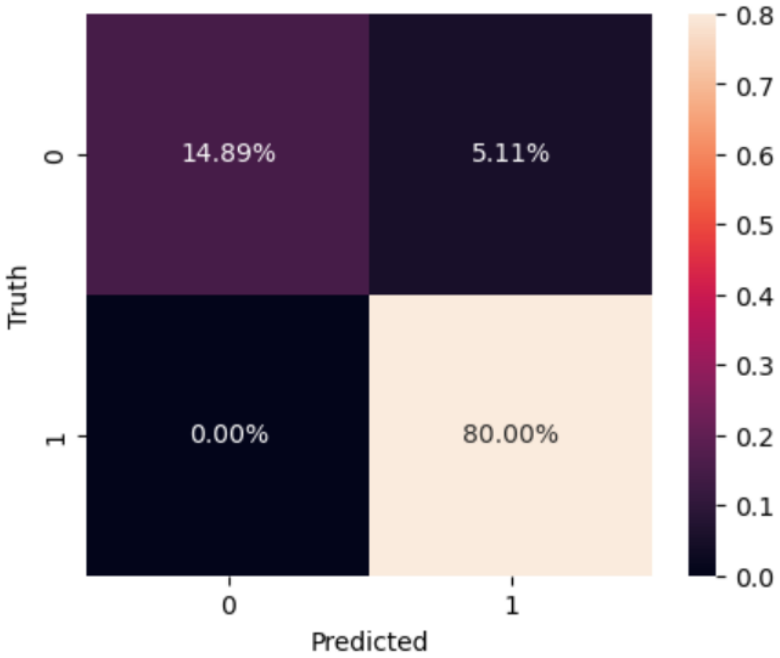
Figure 54: Full covariance matrix for each class and is different

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the full covariance matrix is taken. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 62.90%
Mean Precision: 62.98%
Mean Recall: 62.90%
Mean F1-score on Testing Dataset: 62.85%

Confusion Matrix:

Table 18: Confusion Matrix for Bayes Classifier's Case 4 implemented on Nonlinearly Separable Classes



The decision region plot for both the classes along with the training data superimposed is shown in Figure 55, while the decision region plot for both the classes along with the testing data superimposed in shown in Figure 56.

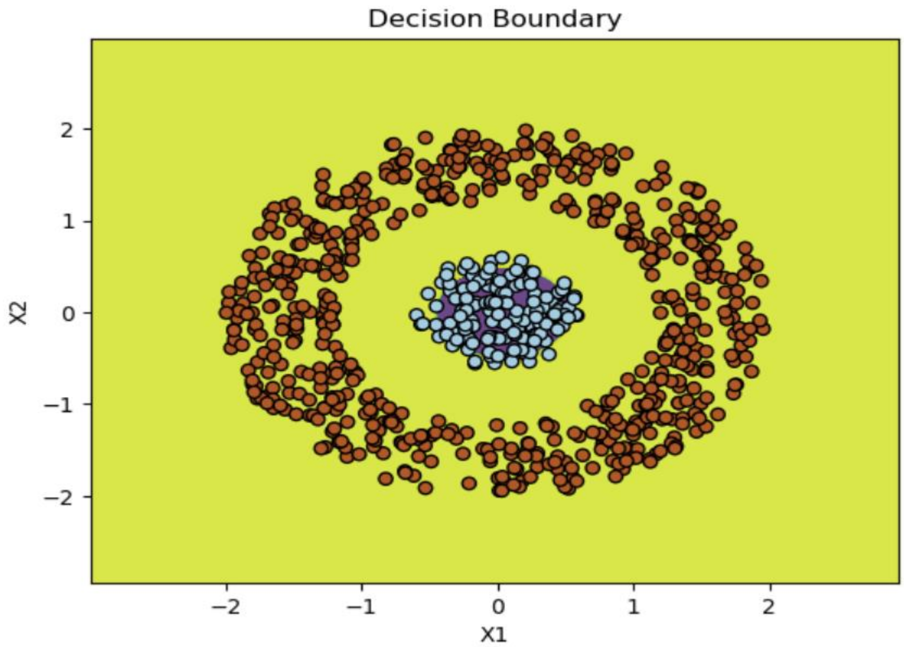


Figure 55: Decision region plot for both the classes superimposed with the training data for Bayes Classifier's Case 4

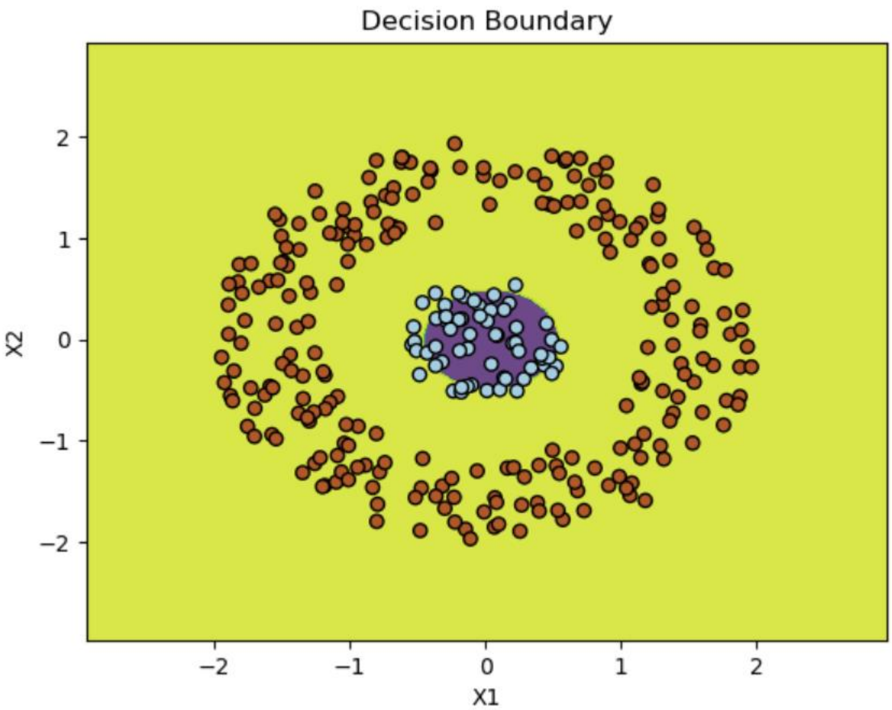


Figure 56: Decision region plot for both the classes superimposed with the testing data for Bayes Classifier's Case 4

As observed from figures 55 and 56, once again, the decision boundaries formed are not quite linear. This is again due to the fact that the covariance matrices for both the classes are different.

3 Overlapping Dataset

Our dataset consists of three overlapping classes. We have labeled the features as Feature 1 and Feature 2, while the class labels corresponding to Class 0 is 0, Class 1 is 1 and Class 2 is 2. The same color scheme has been followed as for the Linearly Separable Classes dataset. The training dataset along with the mean for each class is shown in Figure 68. The figure shows three overlapping classes. As observed from Figure 68, the mean for class 0 is (-0.13, -0.03), class 1 is (3.56, 3.37) and class 2 is (4.91, -1.55).

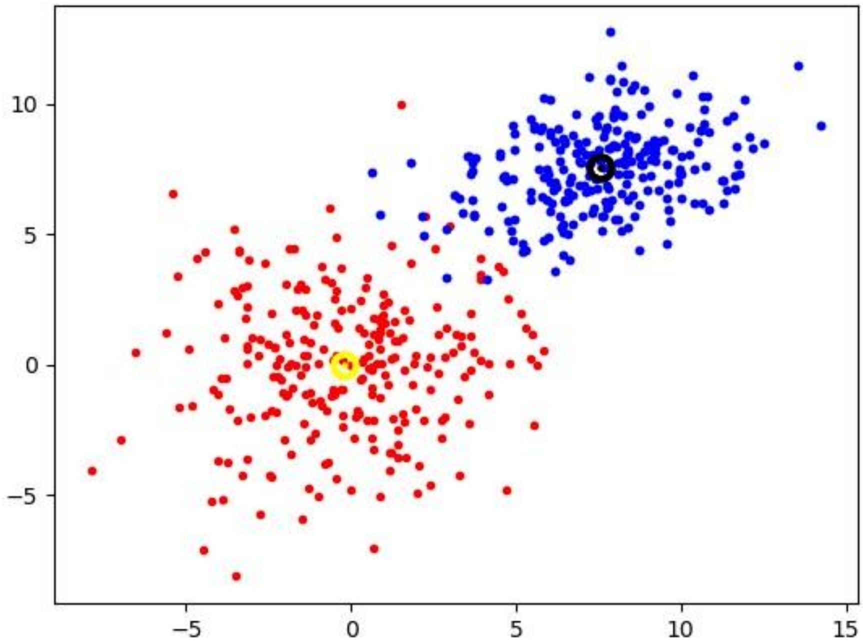


Figure 57: Training data of Overlapping class dataset along with the means of each class

We now try to apply the Nearest Neighbour classifier to see the decision boundaries and accuracy of the classifier.

3.1 Nearest Neighbour Classifier

We have implemented the Nearest Neighbour Classifier in Python using the KNeigboursClassifier model from sklearn. We have set the value of K to be 1 for this case. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Test Set: 84.33%

Accuracy on Validation Set: 88%

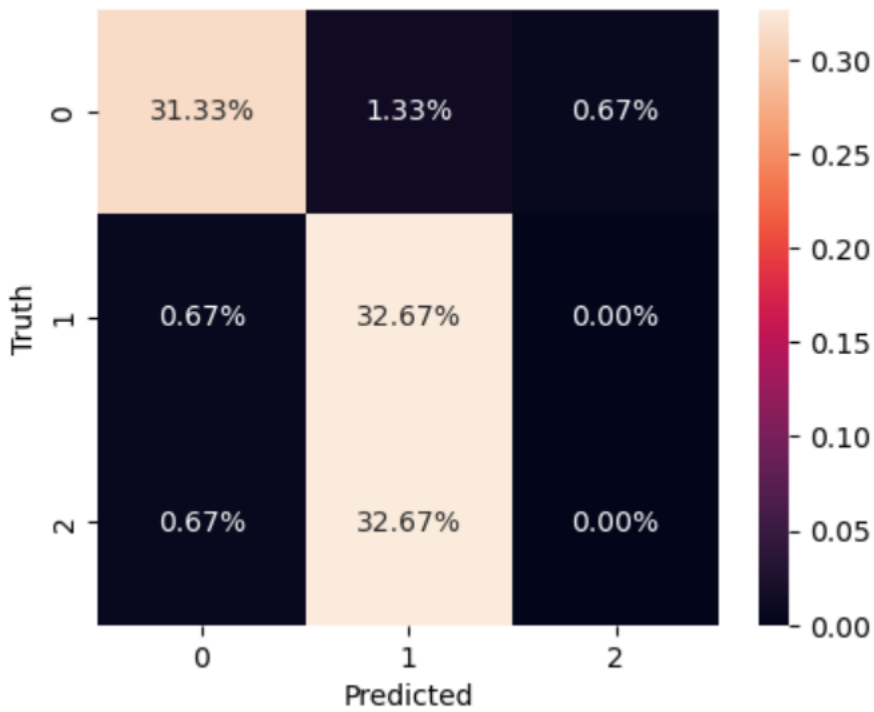
Mean Precision: 84.4477576%

Mean Recall: 84.33333%

Mean F1-score on Testing Dataset: 84.3344223%

Confusion Matrix:

Table 19: Confusion Matrix for Nearest Neighbour Classifier implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 58, while the decision region plot for all classes along with the testing data superimposed in shown in Figure 59. As observed from the figures, a few samples are classified incorrectly, but overall the classifier performs quite well on the data.

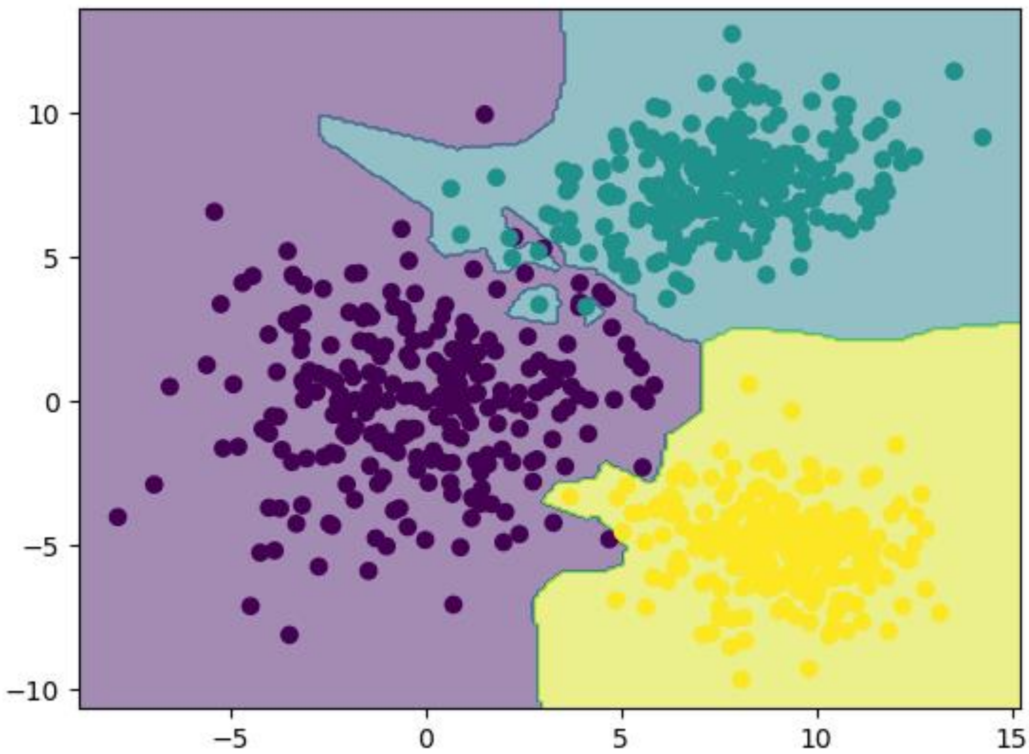


Figure 58: Decision region plot for all the classes along with the training data superimposed as obtained by the Nearest Neighbour Classifier

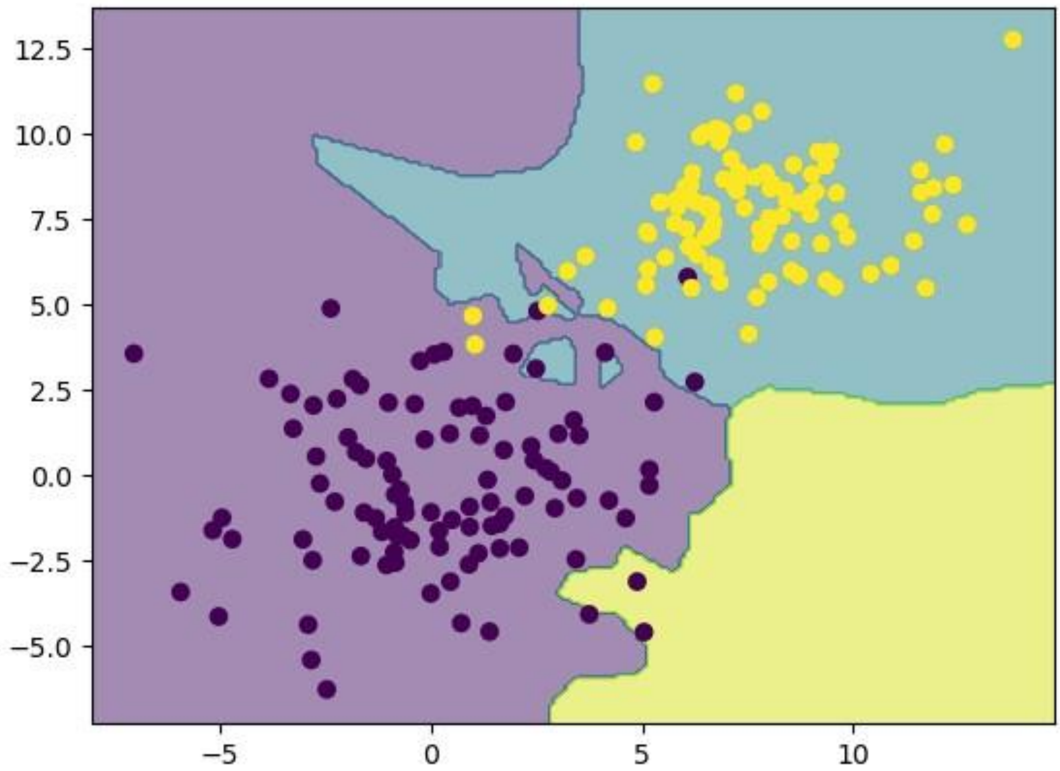


Figure 59: Decision region plot for all the classes along with the testing data superimposed as obtained by the Nearest Neighbour Classifier

3.2 K-Nearest Neighbour Classifier

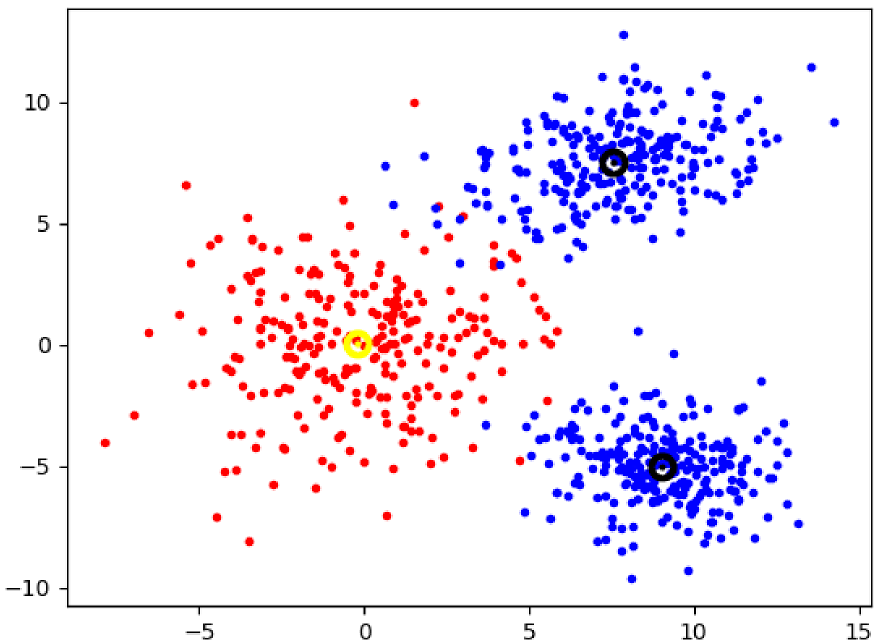


Figure 60: K-Nearest Neighbour Classifier

We have implemented the K- Nearest Neighbour Classifier in Python using the KNeigboursClassifier model from sklearn. In order to find the K value that is optimal for our case, we have fit the classifier for increasing values of K up to half of the length of the training dataset. The plot of K values vs. accuracy is shown below:

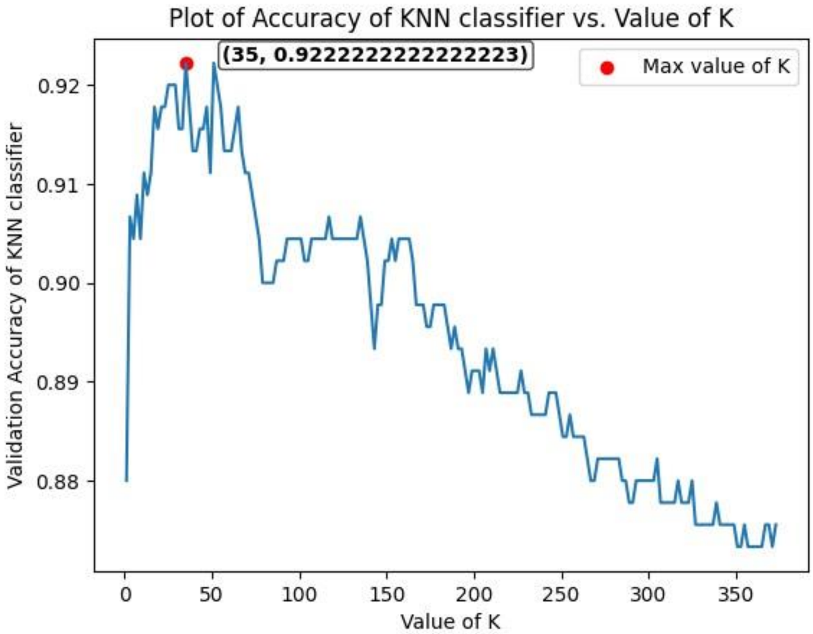


Figure 61: Training Data of Overlapping class dataset along with the means of each class

As observed from Figure 61, the accuracy is maximum at K=35. We have therefore used K=35 to train our classifier. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Test Set: 90%

Accuracy on Validation Set: 92.222%

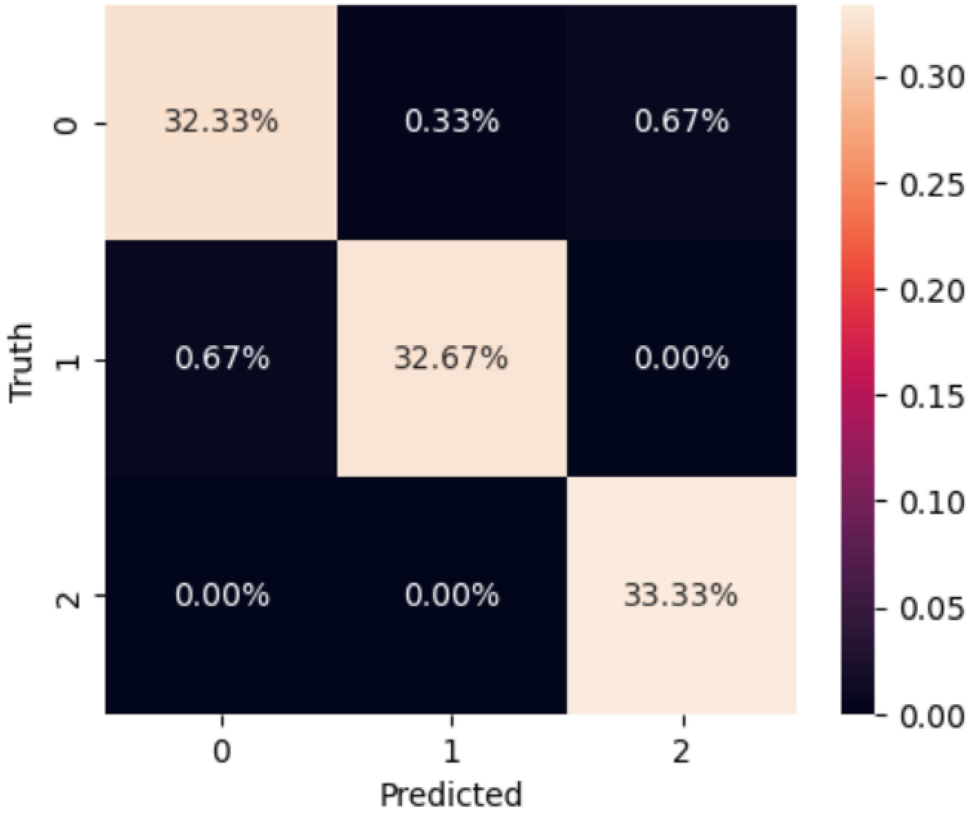
Mean Precision: 90.1621842%

Mean Recall: 90%

Mean F1-score on Testing Dataset: 90.015748%

Confusion Matrix:

Table 20: Confusion Matrix for K-Nearest Neighbour Classifier implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 62, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 63.

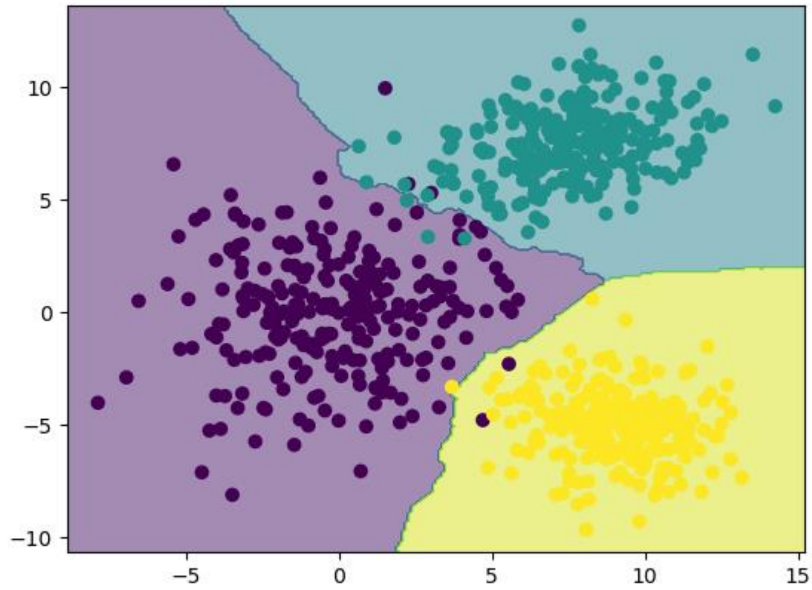


Figure 62: Decision region plot for all the classes along with the training data superimposed as obtained by the K-Nearest Neighbour Classifier

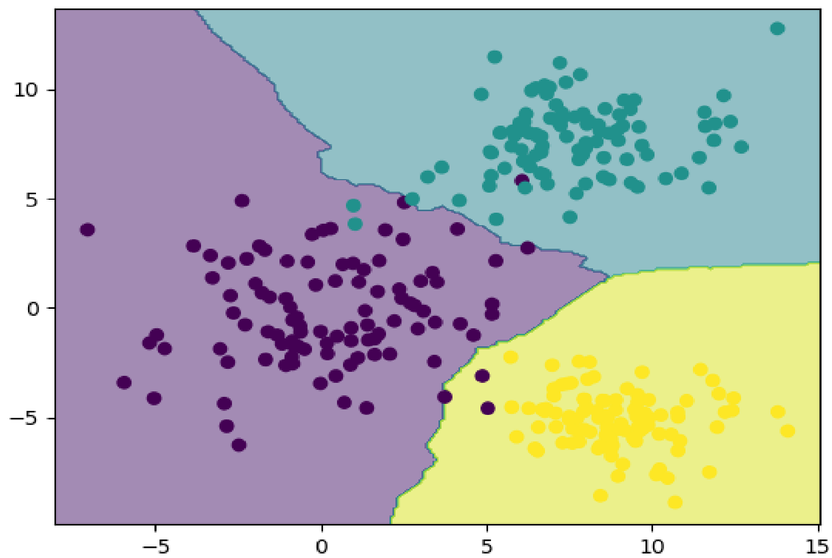


Figure 63: Decision region plot for all the classes along with the testing data superimposed as obtained by the K-Nearest Neighbour Classifier

As observed from the figures, the KNN classifier is a good fit for our overlapping dataset.

3.3 Reference Template-Based Classifier

We have implemented the reference template-based classifier for both sample mean and sample mean and covariance matrix-based classifier.

4.3.1 Mean Vector as Reference Template for a Class

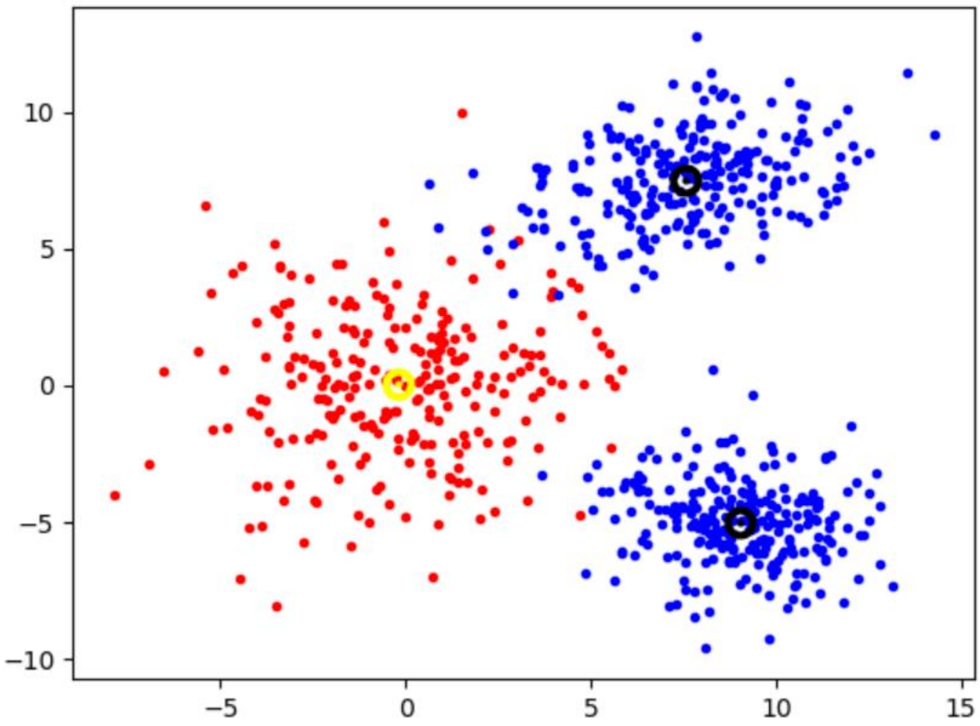


Figure 64: Mean Vector as Reference Template for a Class

We have implemented our own function for the mean vector-based classifier. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Training Set: 85.8478%

Accuracy on Testing Set: 87.290%

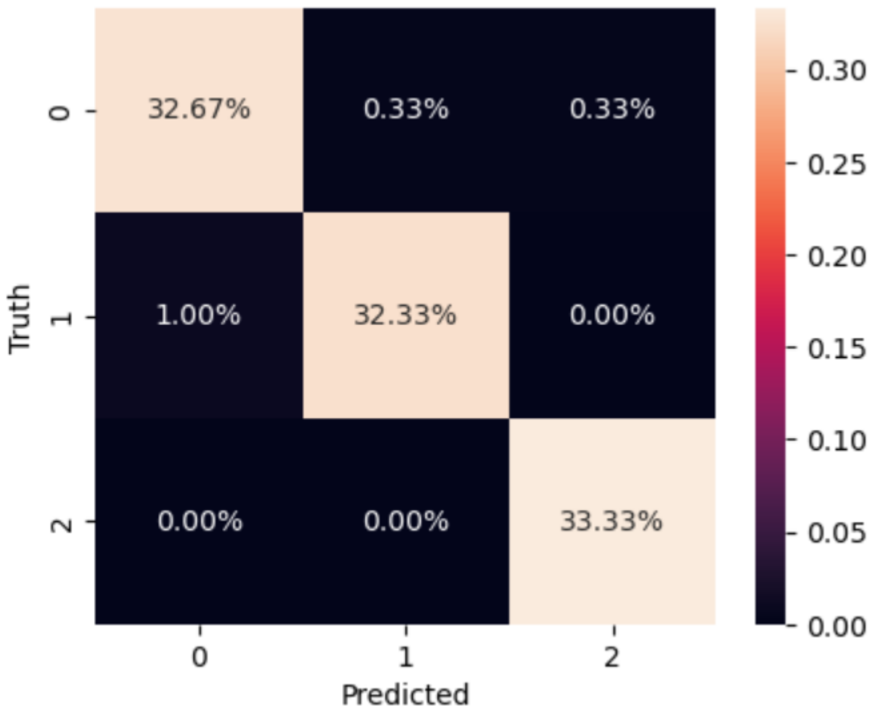
Mean Precision: 87.3737%

Mean Recall: 87.2862%

Mean F1-score on Testing Dataset: 87.27342%

Confusion Matrix:

Table 21: Confusion Matrix for Mean Vector-based Classifier implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 65, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 66.

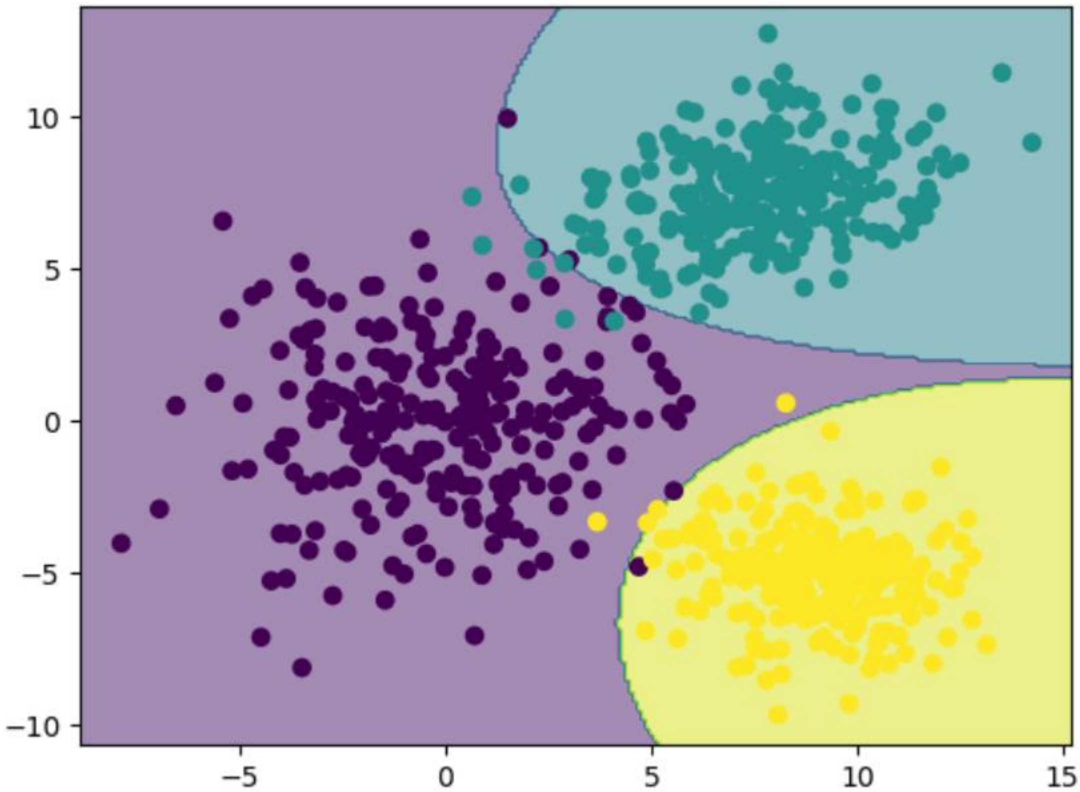


Figure 65: Decision region plot for all the classes along with the training data superimposed as obtained by the Mean vector-based Classifier

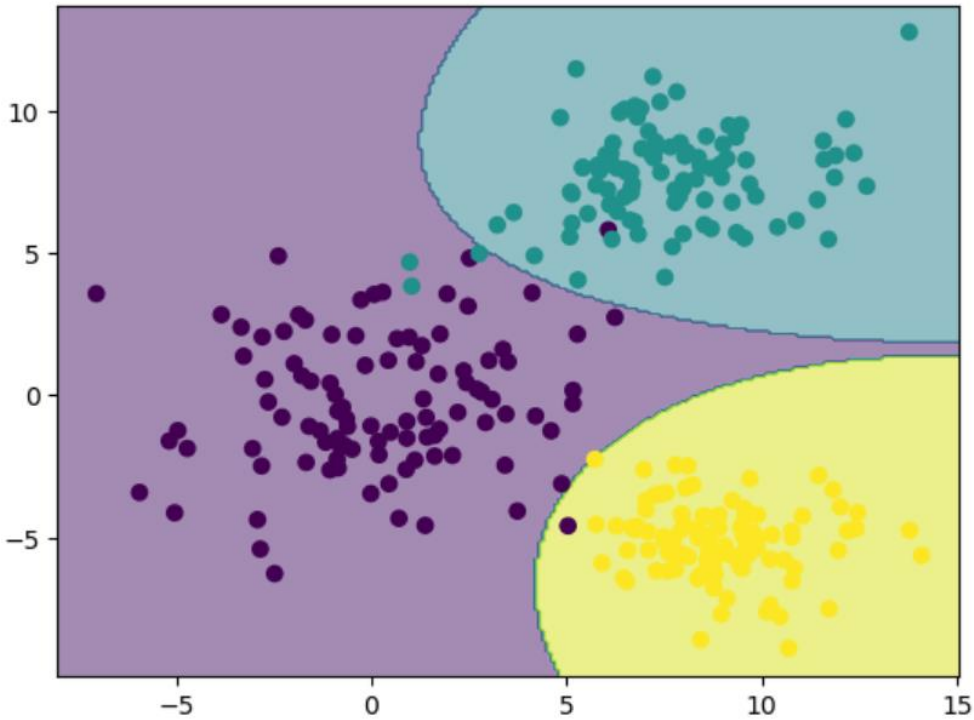


Figure 66: Decision region plot for all the classes along with the testing data superimposed as obtained by the Mean vector-based Classifier

As observed from the figures, the decision boundaries formed are linear. Overall, the classifier is a good fit on the data.

4.3.2 Mean Vector and Covariance Matrix as Reference Template for a Class

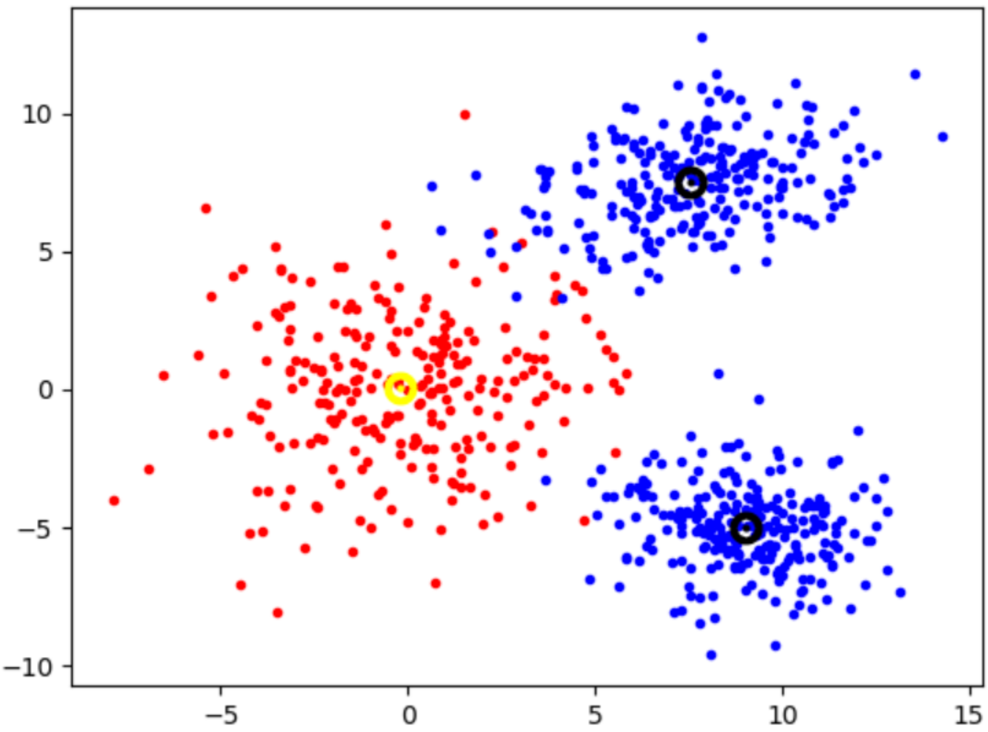


Figure 67: Mean Vector and Covariance Matrix as Reference Template for a Class

We have implemented our own function for the mean vector and covariance matrix-based classifier. The Mahalonobis distance metric was imported from the scipy library. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 75.3333%

Accuracy on Validation Set: 77.1111%

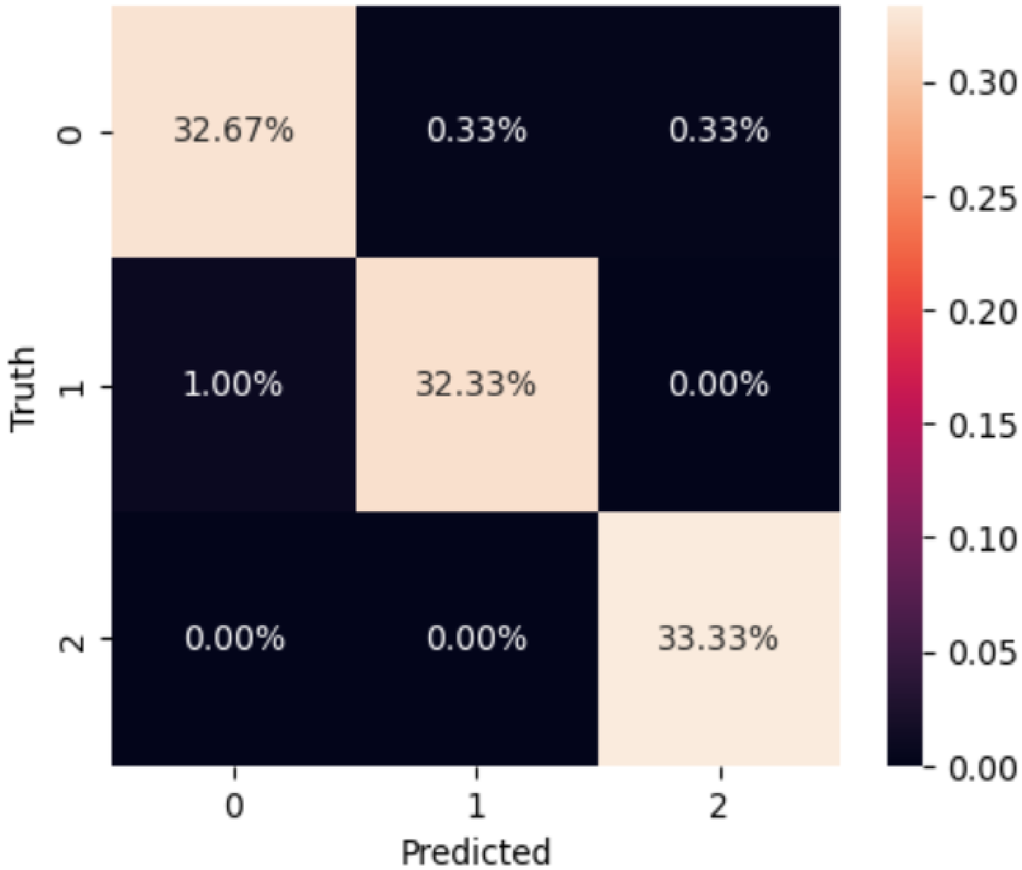
Mean Precision: 75.80243%

Mean Recall: 75.3333%

Mean F1-score on Testing Dataset: 75.09711%

Confusion Matrix:

Table 22: Confusion Matrix for Mean Vector and Covariance Matrix-based Classifier implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 68, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 69.

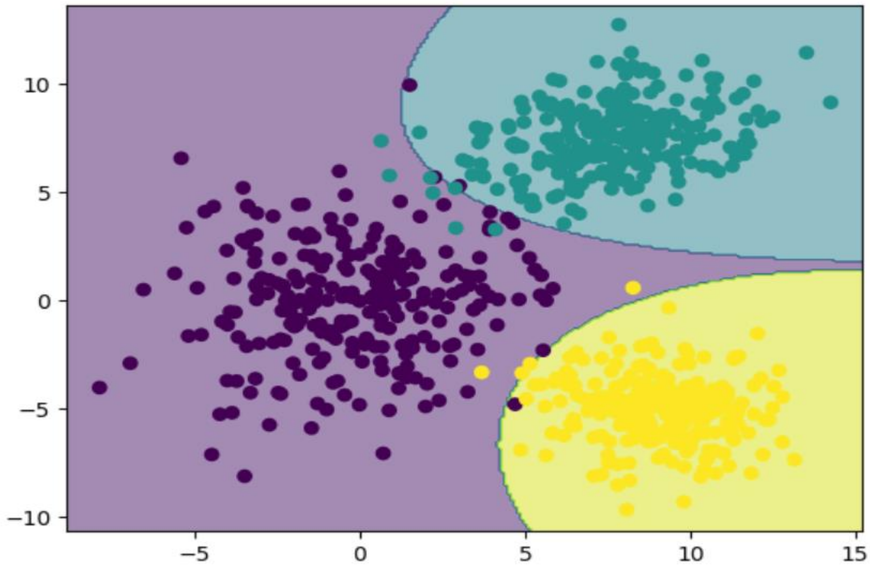


Figure 68: Decision region plot for all the classes along with the training data superimposed as obtained by the Mean vector and Covariance Matrix-based Classifier

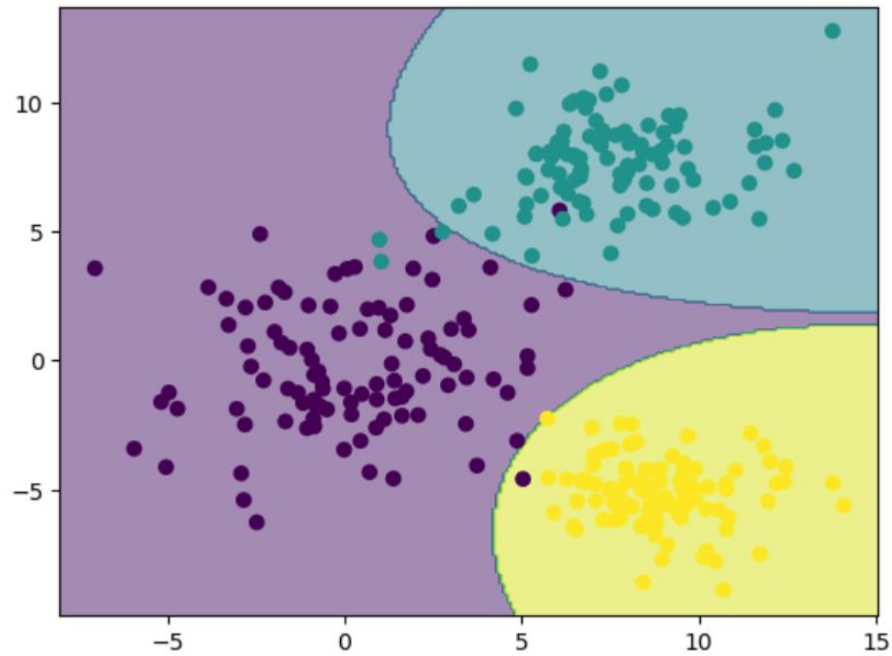


Figure 69: Decision region plot for all the classes along with the testing data superimposed as obtained by the Mean vector and Covariance Matrix-based

Classifier

As observed from the figures, contrary to the Mean Vector-based classifier, the decision boundaries formed are nonlinear.

3.4 Bayes Classifier-Unimodal Gaussian Density

Bayes Classifier was implemented for all four cases of the covariance matrix. The four cases are listed below.

4.4.1 Covariance matrix for all the classes is the same and is $\sigma^2 I$

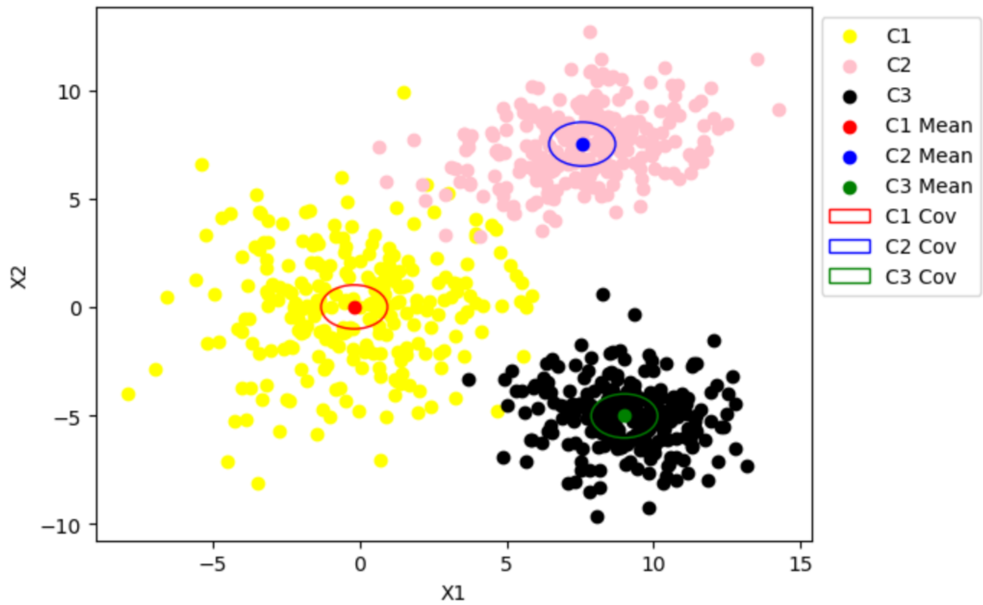


Figure 70: Covariance matrix for all the classes is the same and is σ2I

The features are independent in this case and both the features have the same variance. The Gaussian density function was implemented using the *multivariate _normal* module from SciPy.stats. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 87.29097%

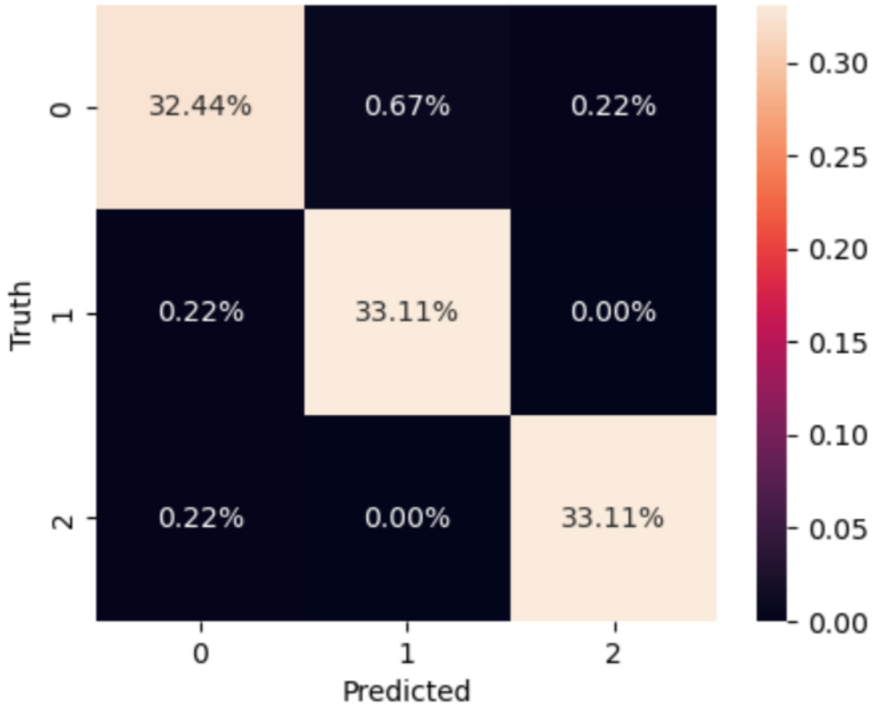
Mean Precision: 87.519889%

Mean Recall: 87.27609%

Mean F1-score on Testing Dataset: 87.26854%

Confusion Matrix:

Table 23: Confusion Matrix for Bayes Classifier's Case 1 implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 71, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 72.

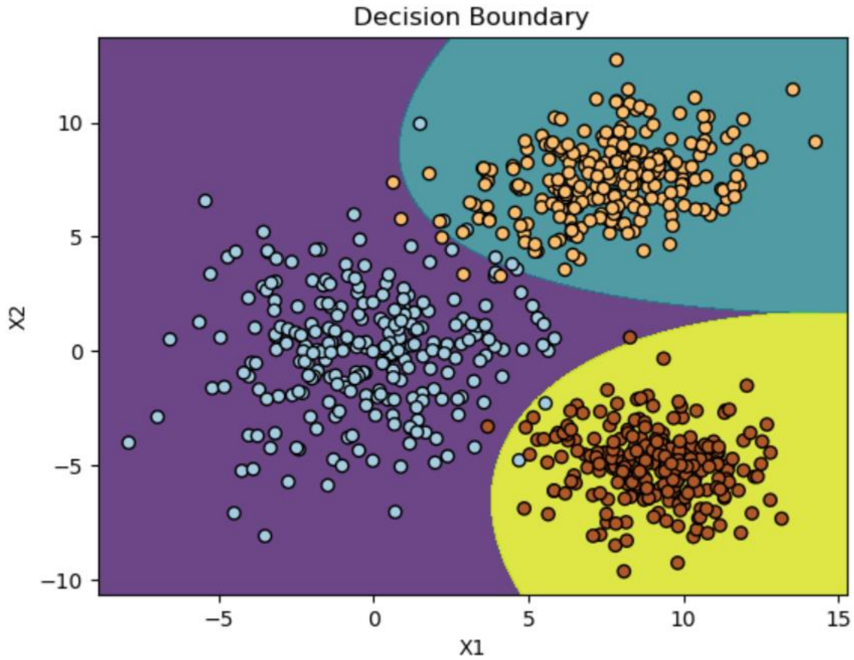


Figure 71: Decision region plot for all the classes along with the training data superimposed as obtained by the Bayes Classifier's Case 1

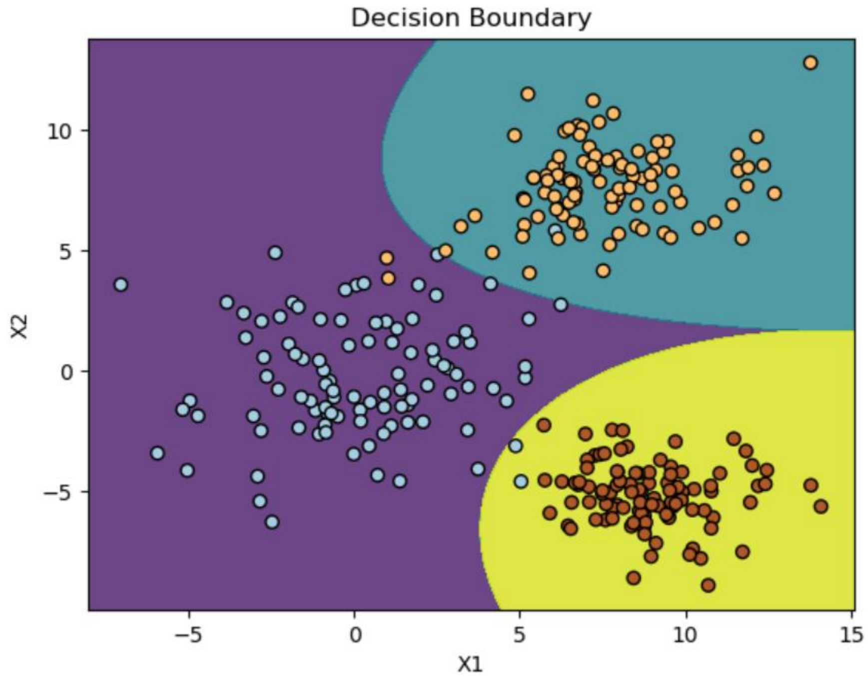


Figure 72: Decision region plot for all the classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 1

As observed from the figures, Bayes Classifier's Case 1 forms linear decision boundaries.

4.4.2 Full covariance matrix for all the classes and is same for all the classes - Part 1

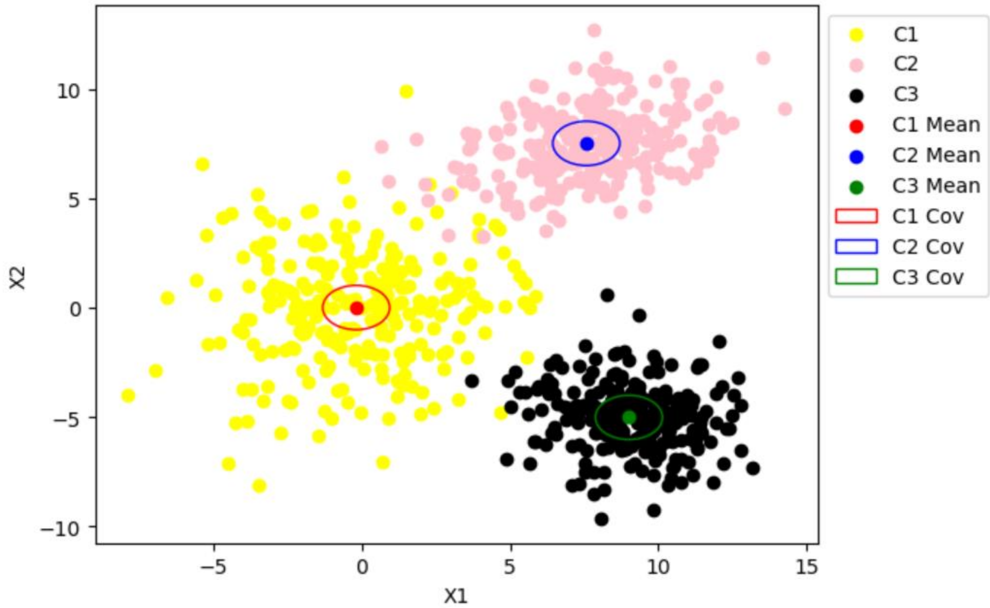


Figure 73: Full covariance matrix for all the classes and is same for all the classes - Part 1

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by taking the average of covariance matrices of all the classes. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 87.62542%

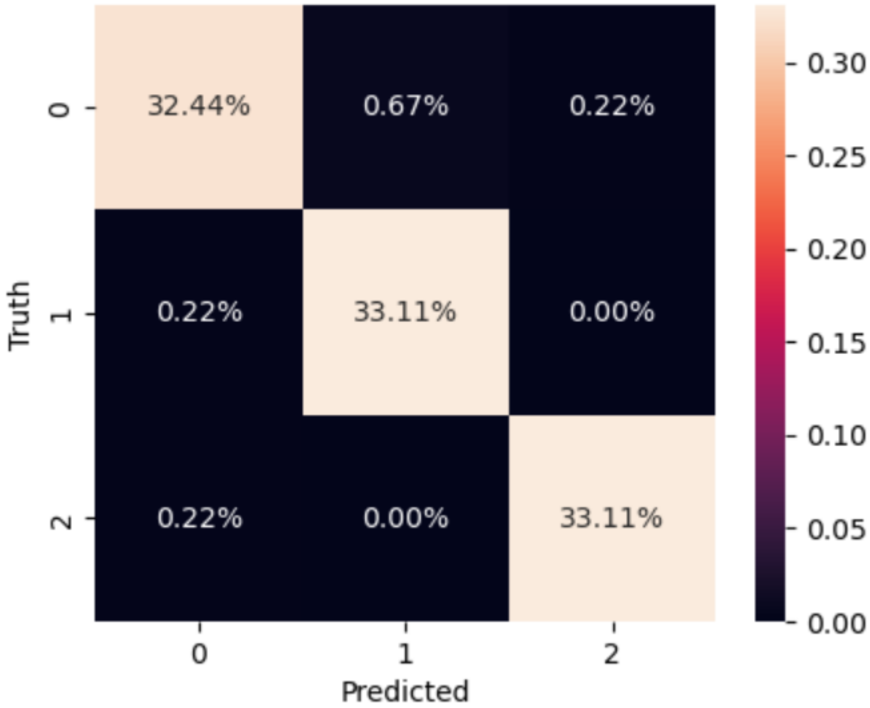
Mean Precision: 87.817%

Mean Recall: 87.6128%

Mean F1-score on Testing Dataset: 87.611425%

Confusion Matrix:

Table 24: Confusion Matrix for Bayes Classifier's Case 2 Part 1 implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 74, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 75.

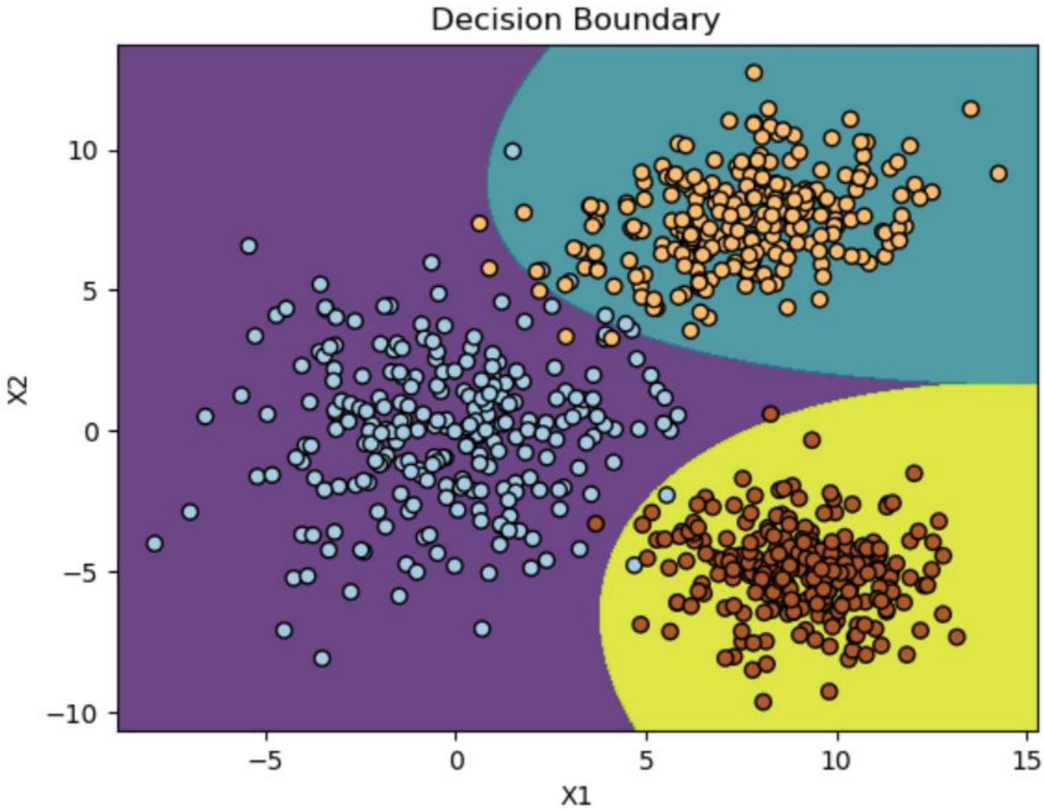


Figure 74: Decision region plot for all the classes along with the training data superimposed as obtained by the Bayes Classifier's Case 2 Part 1

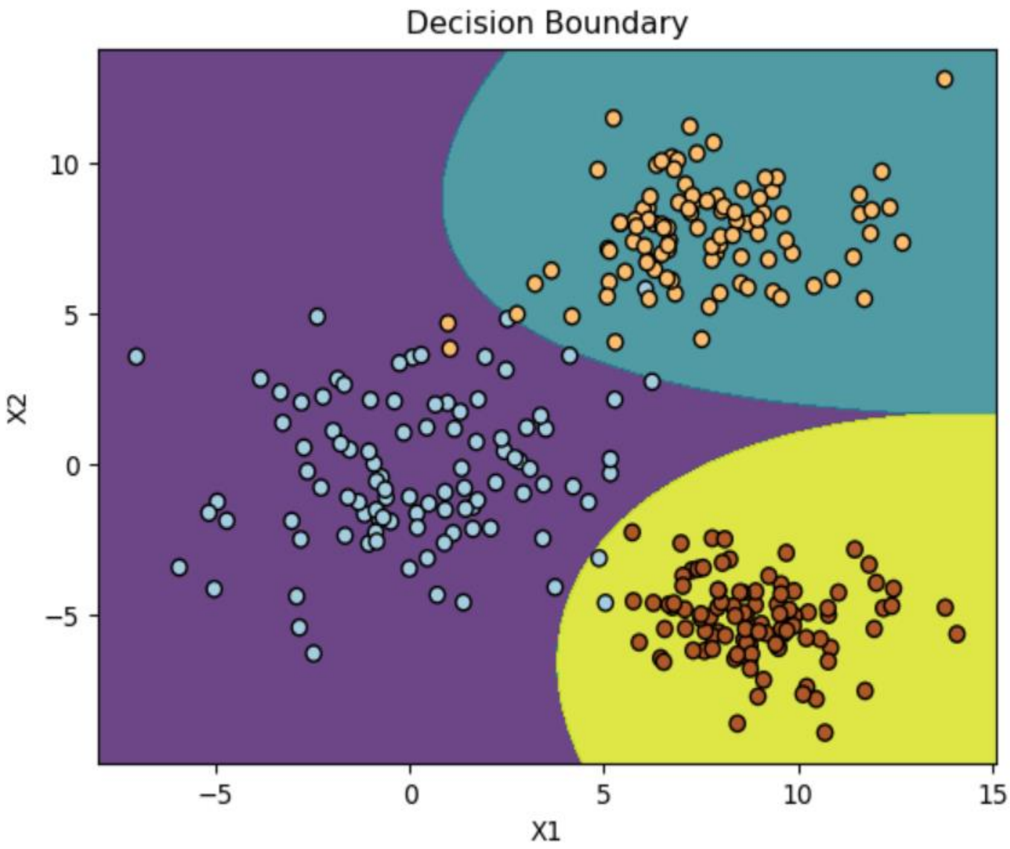


Figure 75: Decision region plot for all the classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 2 Part 1

As observed from the figures, the decision boundaries for this case are also linear.

4.4.3 Full covariance matrix for all the classes and is same for all the classes - Part 2

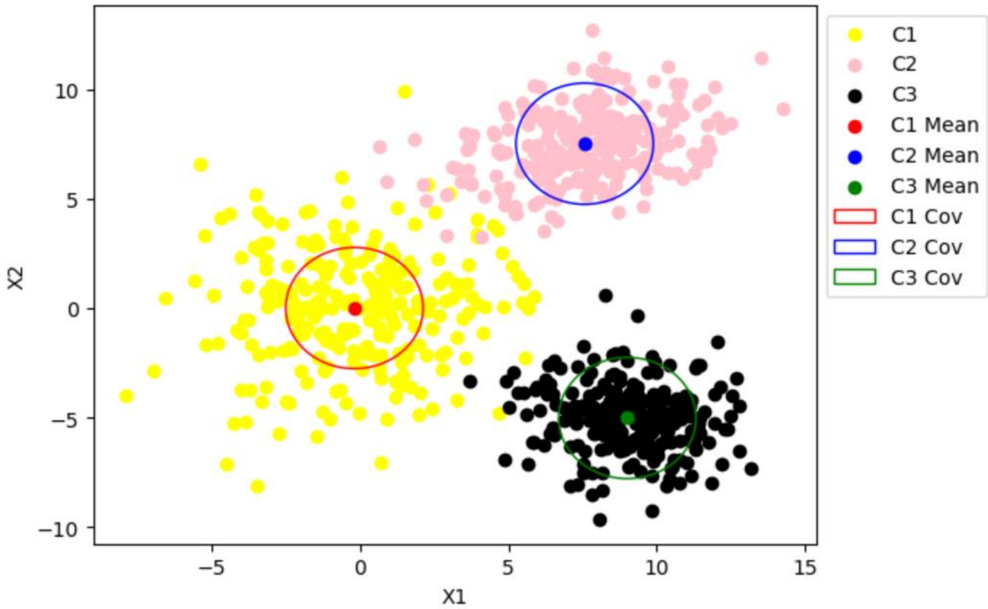


Figure 76: Full covariance matrix for all the classes and is same for all the classes - Part 2

The features are dependent in this case and the variance for each feature is different. Both the classes have the same covariance matrix, which has been calculated by computing the covariance matrix of training data of all the classes combined. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 87.29097%

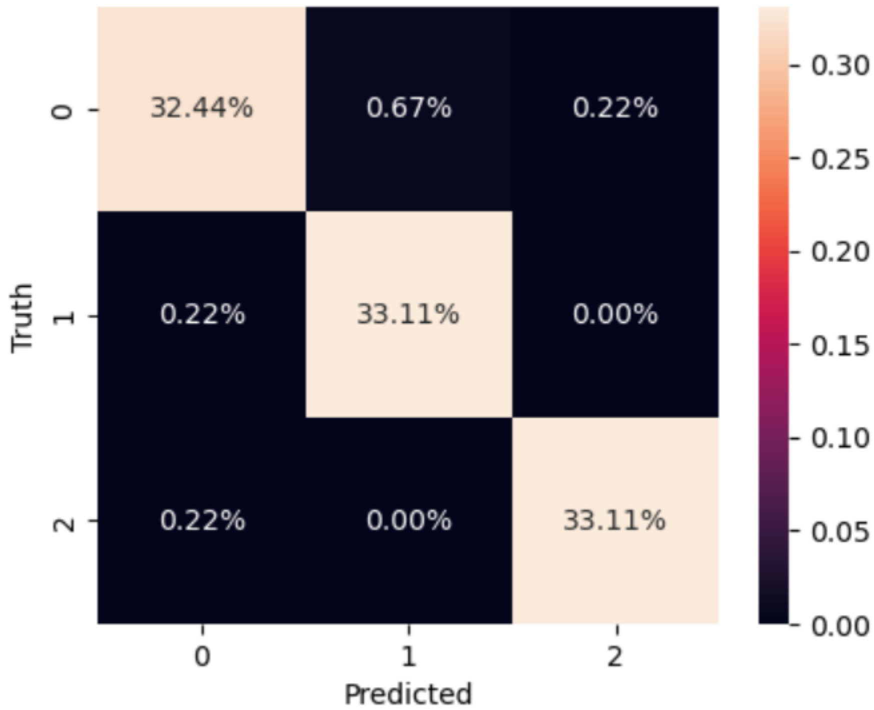
Mean Precision: 87.43997%

Mean Recall: 87.2795%

Mean F1-score on Testing Dataset: 87.2638%

Confusion Matrix:

Table 25: Confusion Matrix for Bayes Classifier's Case 2 Part 2 implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 77, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 78.

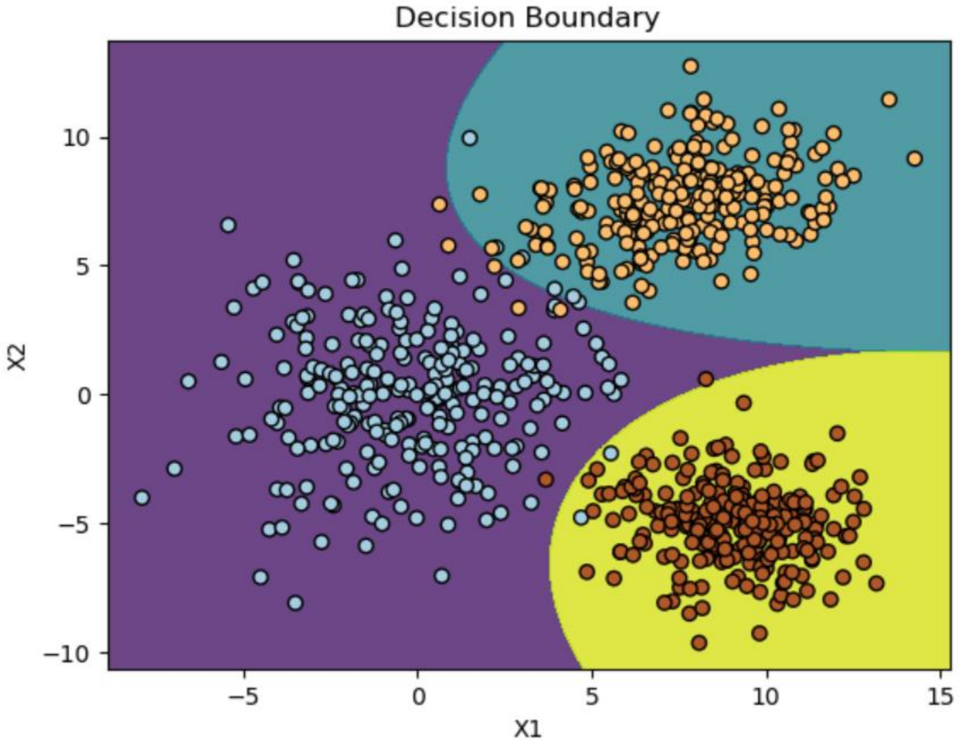


Figure 77: Decision region plot for all the classes along with the training data superimposed as obtained by the Bayes Classifier's Case 2 Part 2

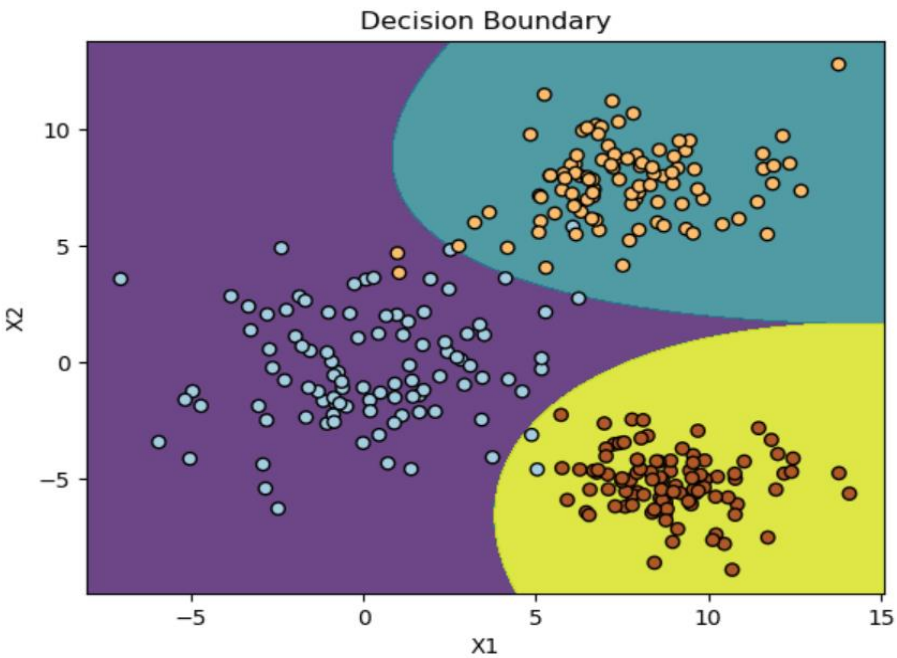


Figure 78: Decision region plot for all the classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 2 Part 2

As observed from the figures above, the decision boundaries formed in this case are also linear.

4.4.4 Covariance matrix is diagonal and is different for each class

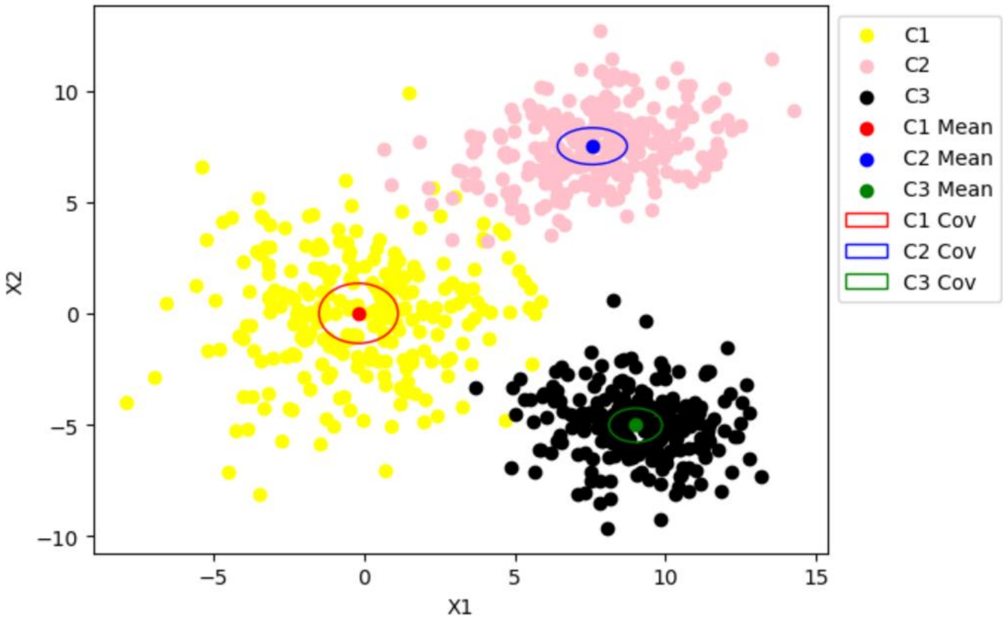


Figure 79: Covariance matrix is diagonal and is different for each class

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the offdiagonal elements for each covariance matrix have been made 0. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 88.96321%

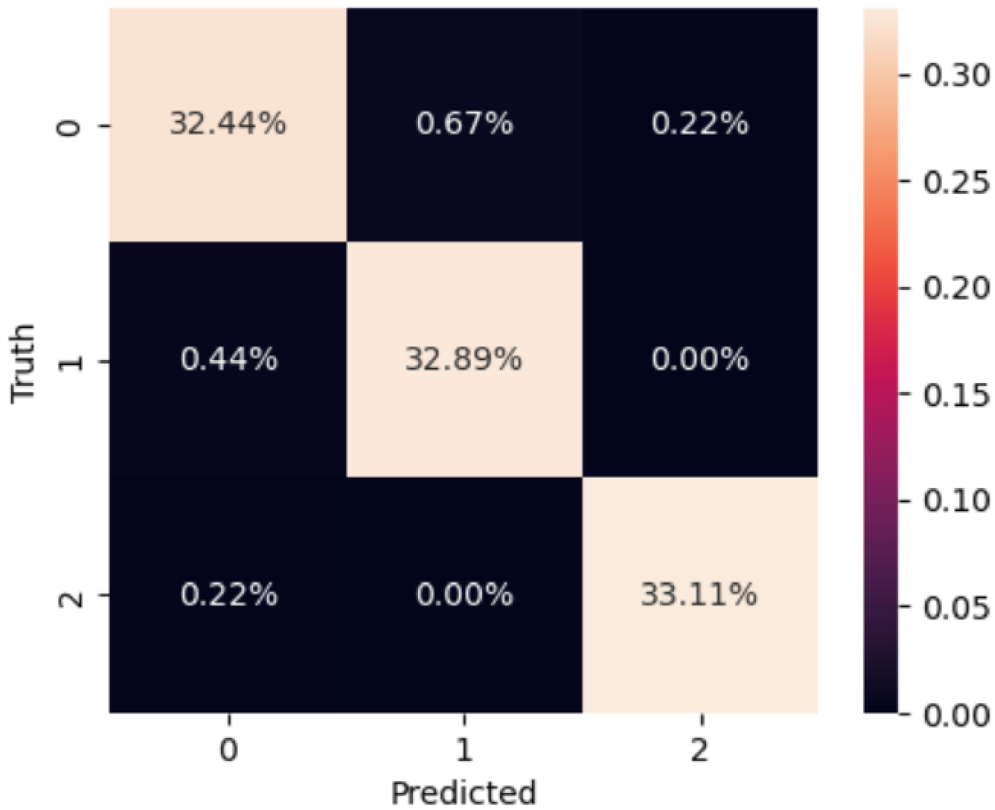
Mean Precision: 89.2661%

Mean Recall: 88.9394%

Mean F1-score on Testing Dataset: 88.9202%

Confusion Matrix:

Table 26: Confusion Matrix for Bayes Classifier's Case 3 implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 80 while the decision region plot for all classes along with the testing data superimposed is shown in Figure 81.

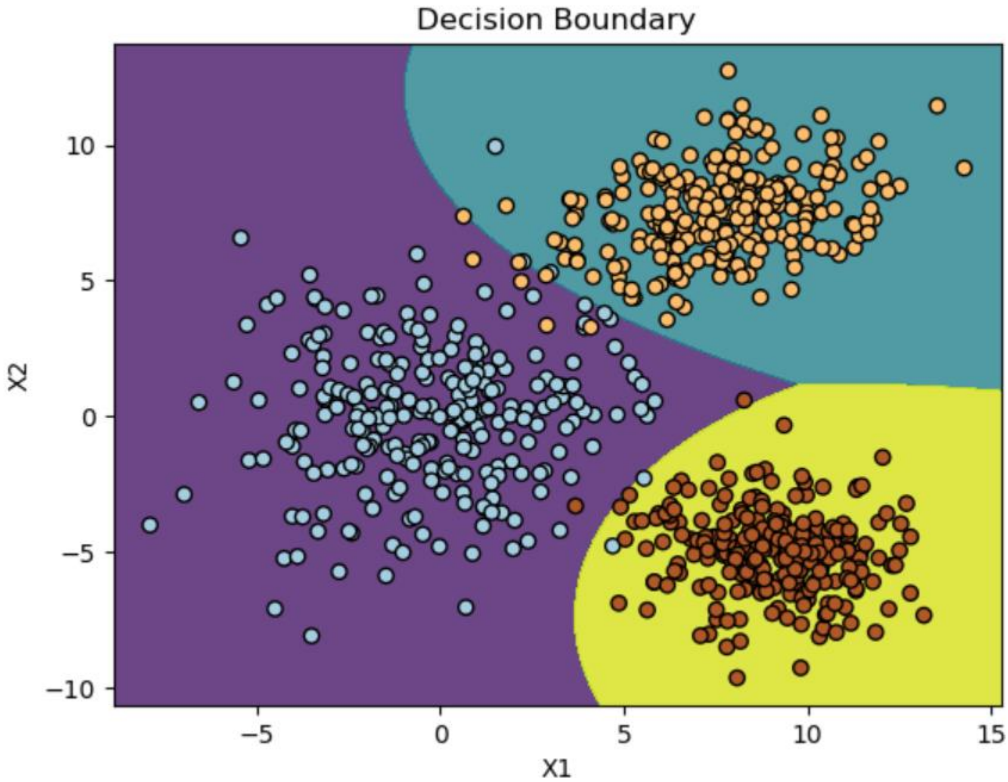


Figure 80: Decision region plot for all the classes along with the training data superimposed as obtained by the Bayes Classifier's Case 3

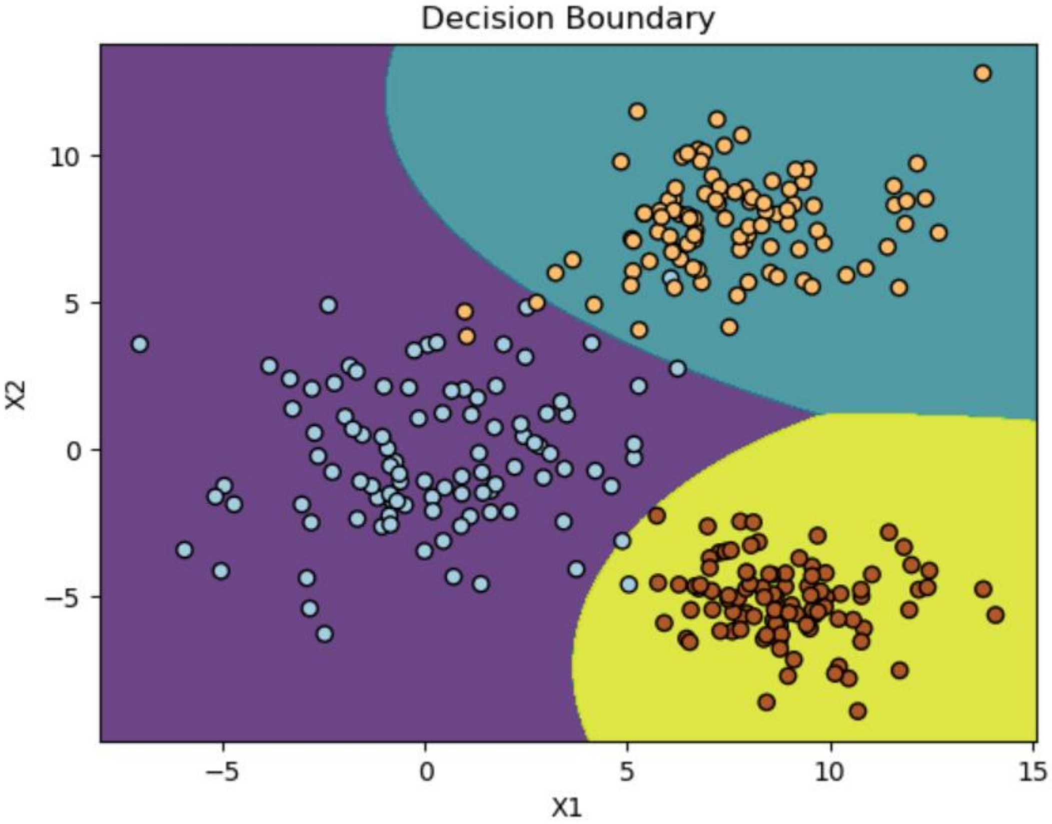


Figure 81: Decision region plot for all the classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 3

As observed from the figures above, contrary to the previous case, the decision boundaries formed are nonlinear. This is because the covariance matrices for all the classes are different.

4.4.5 Full covariance matrix for each class and is different

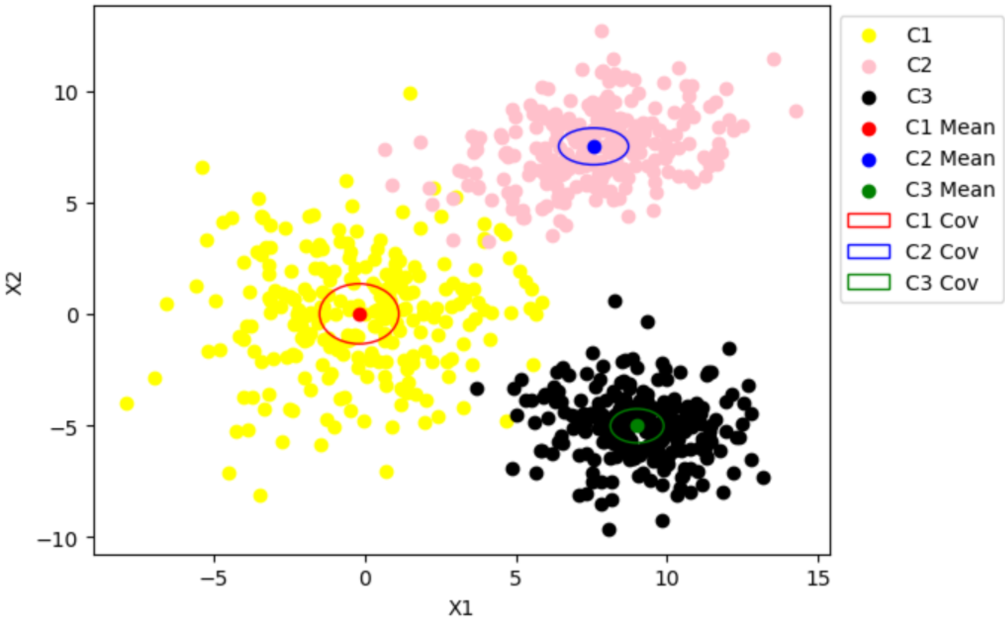


Figure 82: Full covariance matrix for each class and is different

The features are dependent in this case and the variance for each feature is different. The covariance matrices for each class is different and the full covariance matrix is taken. The classifier's performance on the dataset was evaluated using the following performance measures:

Accuracy on Testing Set: 90.30100%

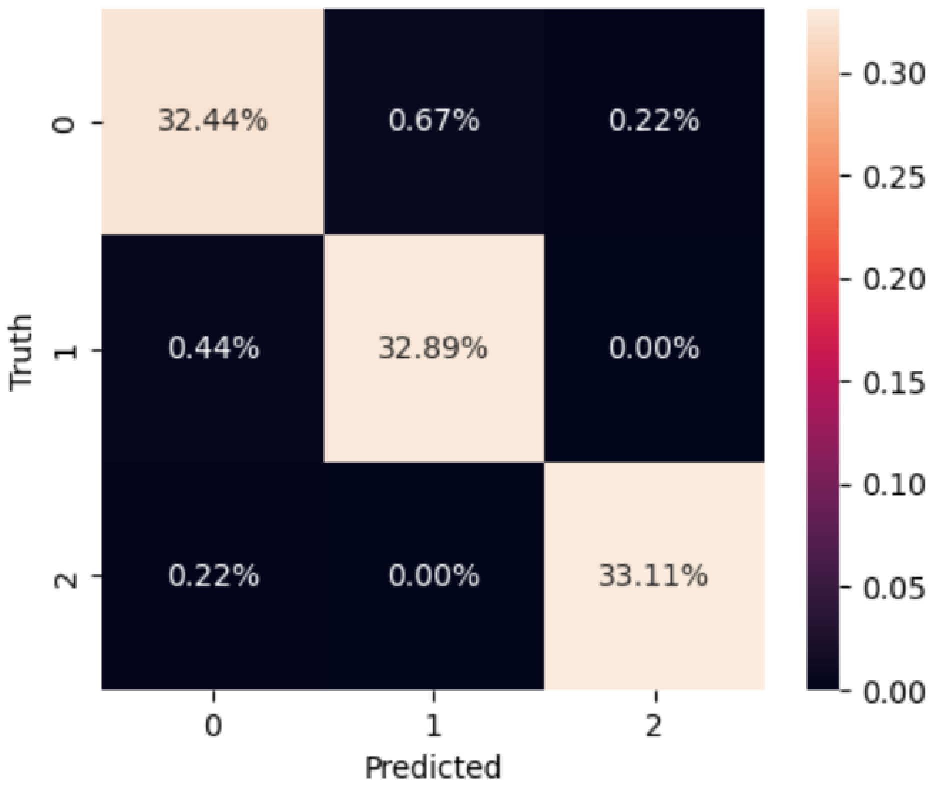
Mean Precision: 90.359000%

Mean Recall: 90.2929%

Mean F1-score on Testing Dataset: 90.287586%

Confusion Matrix:

Table 27: Confusion Matrix for Bayes Classifier's Case 4 implemented on Overlapping Classes



The decision region plot for all classes along with the training data superimposed is shown in Figure 83, while the decision region plot for all classes along with the testing data superimposed is shown in Figure 84.

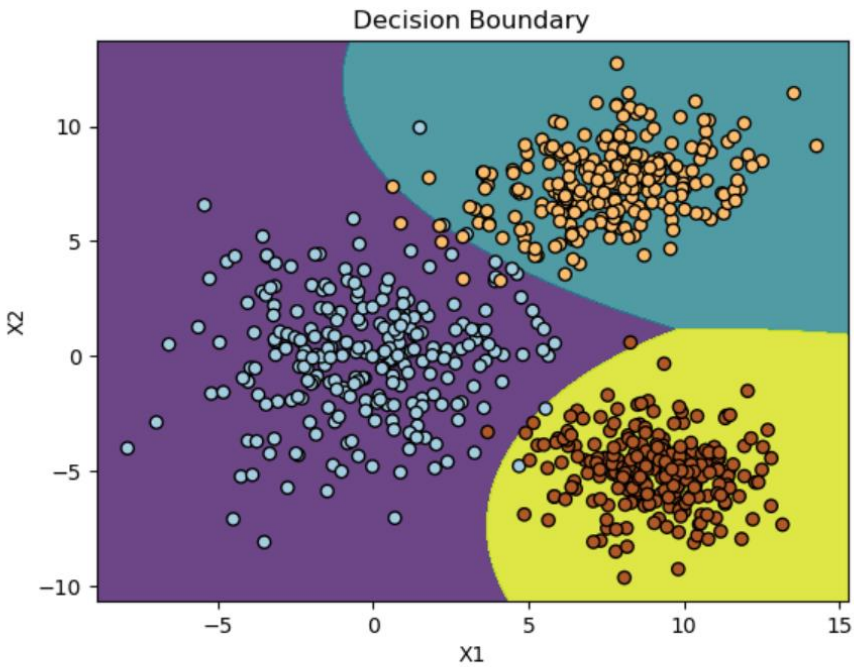


Figure 83: Decision region plot for all the classes along with the training data superimposed as obtained by the Bayes Classifier's Case 4

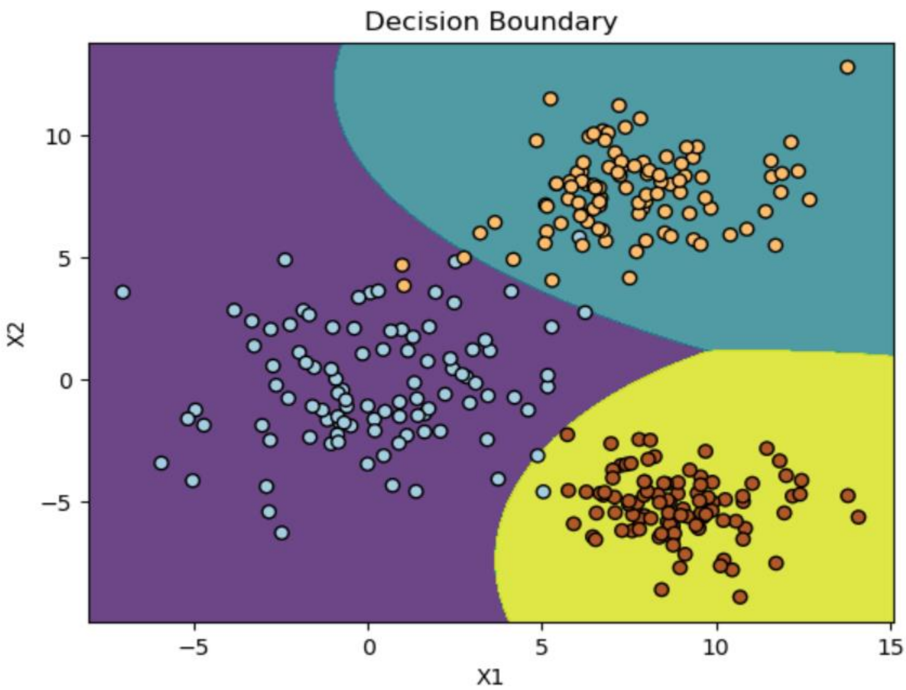


Figure 84: Decision region plot for all the classes along with the testing data superimposed as obtained by the Bayes Classifier's Case 4

4 Observations

The testing set accuracies (in %) for each classifier on each dataset have been summarised in Table 28 below.

Table 28: Testing set accuracies for each classifier

Type of Classifier	**Linearly Separable Classes**	**Non-Linearly Separable Classes**	**Overlapping Classes**
Nearest Neighbor	100	100	84.33
K-Nearest Neighbor	100	100	90
Mean Vector	99.732	63.52	85.84
Mean Vector & Covariance Matrix	97	63.25	75.33
Bayes Classifier Case 1	100	63.114	87.29
Bayes Classifier Case 2 Part 1	100	63.114	87.62
Bayes Classifier Case 2 Part 2	100	63.114	87.29
Bayes Classifier Case 3	100	63.42	88.96

Bayes Classifier Case 4	100	62.90	90.3

The nature of the decision boundary for each classifier on each dataset has been summarised in Table 29 below.

Table 29: Nature of decision boundary for each classifier on each dataset

Type of Classifier	**Linearly Separable Classes**	**Non-Linearly Separable Classes**	**Overlapping Classes**
Nearest Neighbor	Nonlinear	Nonlinear	Nonlinear
K-Nearest Neighbor	Nonlinear	Nonlinear	Nonlinear
Mean Vector	Linear	Linear	Linear
Mean Vector & Covariance Matrix	Linear	Linear	Nonlinear
Bayes Classifier Case 1	Linear	Linear	Linear
Bayes Classifier Case 2 Part 1	Linear	Linear	Linear
Bayes Classifier Case 2 Part 2	Linear	Linear	Linear
Bayes Classifier Case 3	Nonlinear	Nonlinear	Nonlinear
Bayes Classifier Case 4	Nonlinear	Nonlinear	Nonlinear

5 Inferences

Referring Table 28 and 29, the following inferences can be drawn from this project:

1. The Nearest Neighbour classifier performs well on all datasets. However, there is a trade-off between the performance and the efficiency, since the Nearest Neighbour Classification algorithm is computationally expensive, and as observed from the code, the finer the mesh grid on which the datapoints are classified, the longer the code will take to run.

2. A similar inference can be drawn for the K-Nearest Neighbour Classifier. It is also computationally expensive, but as seen from the tables, it gives us a better accuracy for overlapping data.

3. The mean vector-based classifier fits well on linearly separable data, but on data as complex as our nonlinearly separable classes (which are spiral in nature), it fails to form a nonlinear decision boundary that will fit well on our nonlinear data.

4. A similar inference can be drawn for the mean vector and covariance matrix-based classifier. However, this classifier is more computationally expensive as we would also have to evaluate the Mahalonobis distance metric, and also gives a lower performance as compared to the mean vector-based classifier.

5. The Bayesian Classifier's Case 1, in which we consider the features to be independent and to have the same variance σ^2 with a diagonal covariance matrix performs well on linearly separable data as well as overlapping data, but performs poorly on nonlinearly separable data as complex as our dataset since the features are dependent on each other.

6. The Bayesian Classifier's Case 2 Part 1, in which we assume both the features to have the same covariance matrix, which is the average of covariance matrices of all classes, performs similarly to the previous case. The same inferences can be drawn for Case 2 Part 2, Case 3 as well as for Case 4.

7. The Bayesian Classifier's Case 4, in which we take the covariance matrices of the classes to be different, performs better than the other cases for the overlapping data. This may be due to the reason that different covariance matrices imply dependent features, and the features are dependent for the overlapping classes.