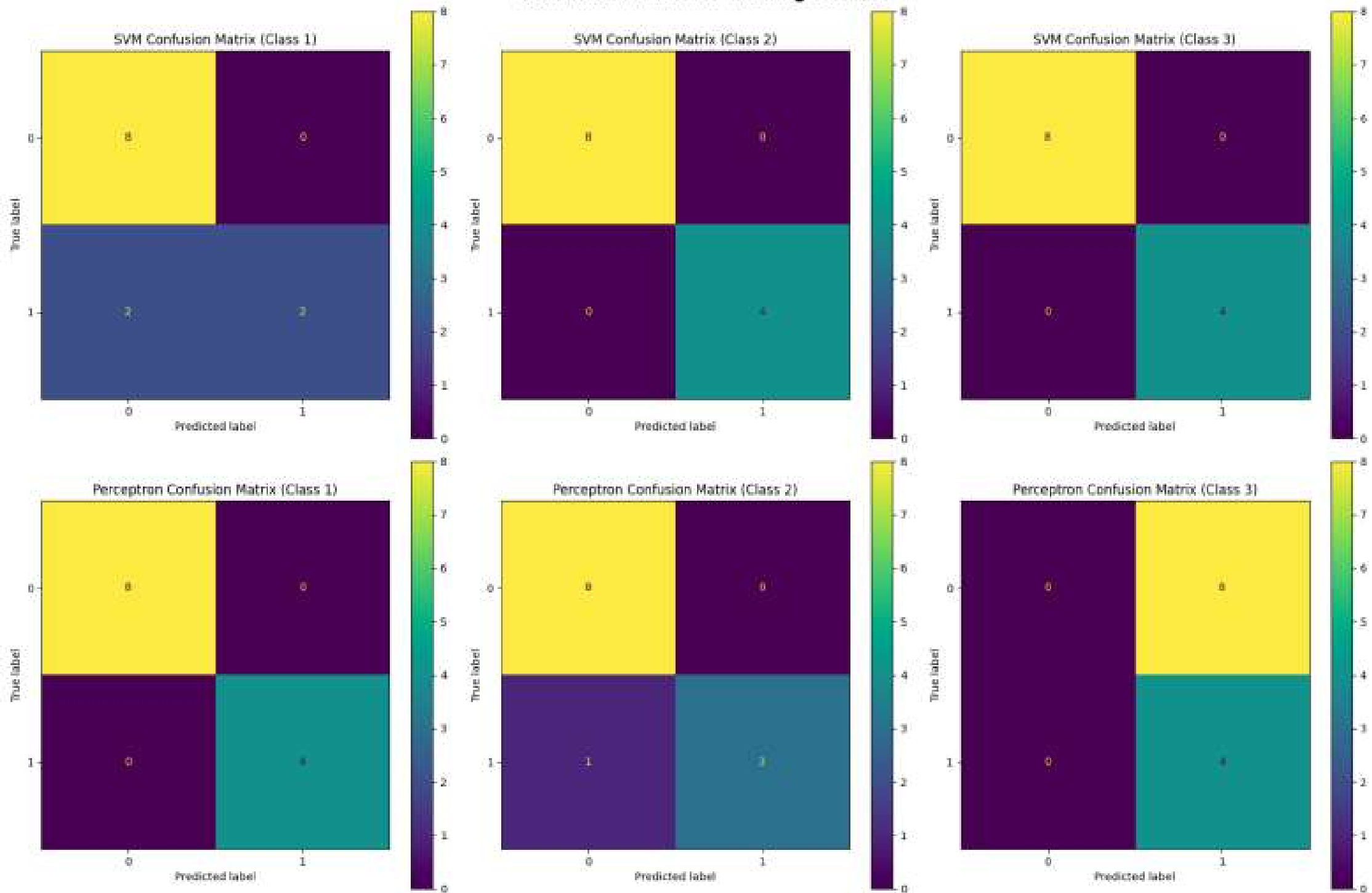
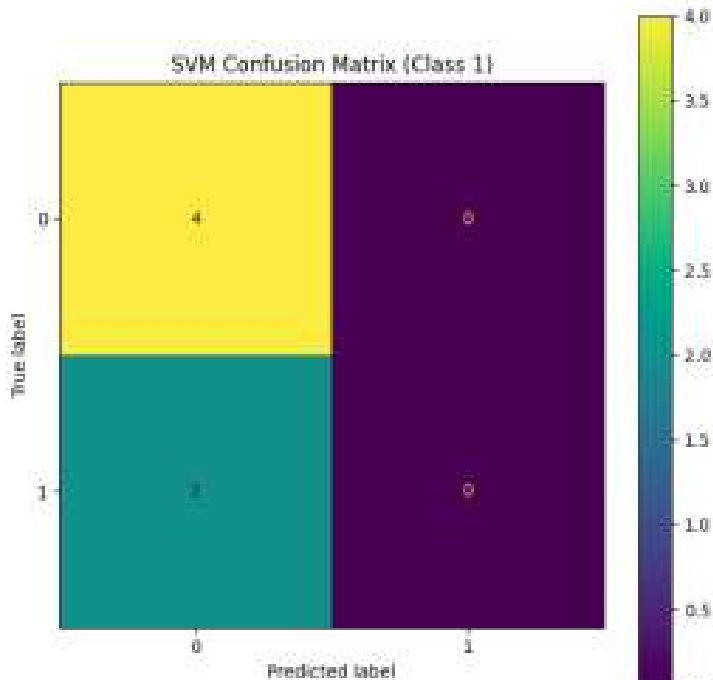


Confusion Matrix for Training Dataset

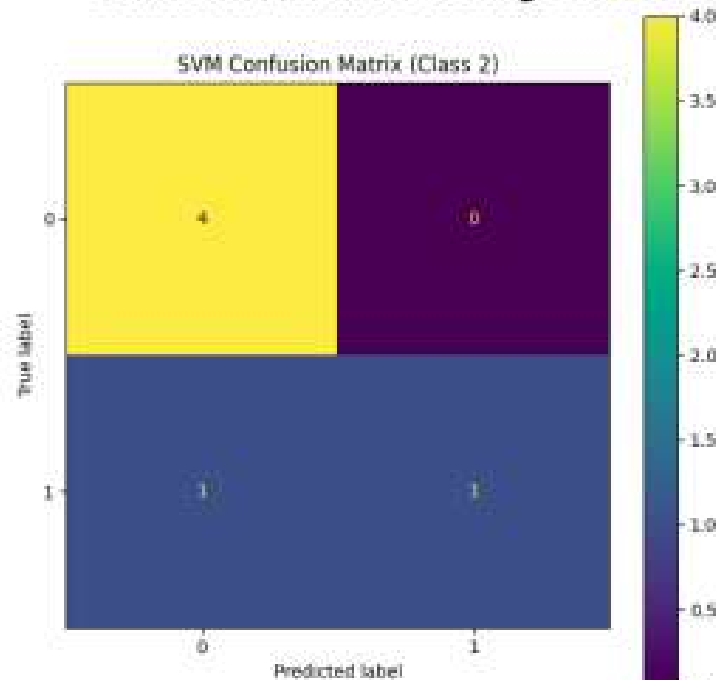


Confusion Matrix for Testing Dataset

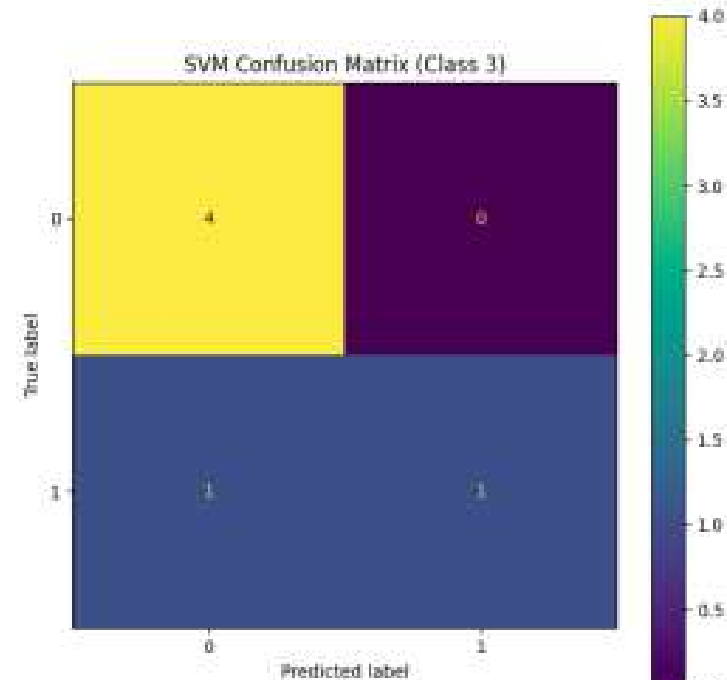
SVM Confusion Matrix (Class 1)



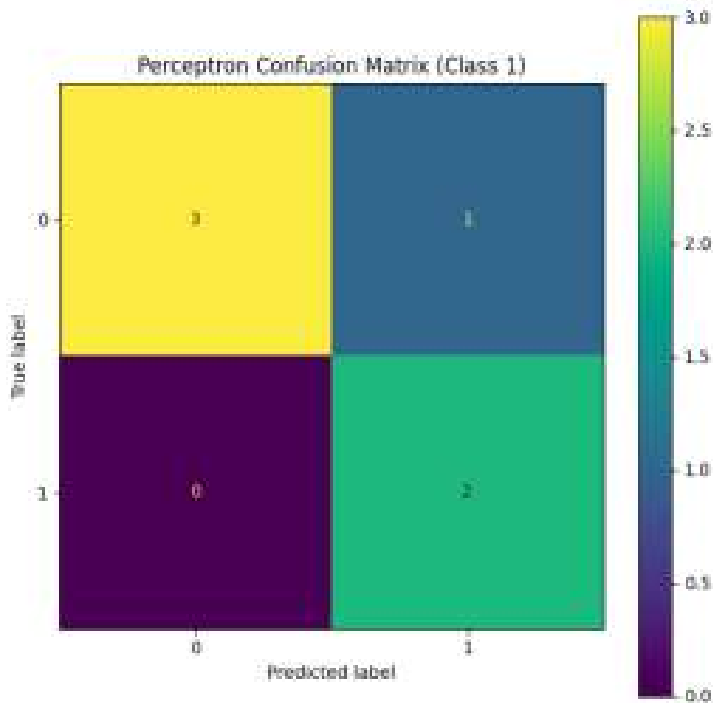
SVM Confusion Matrix (Class 2)



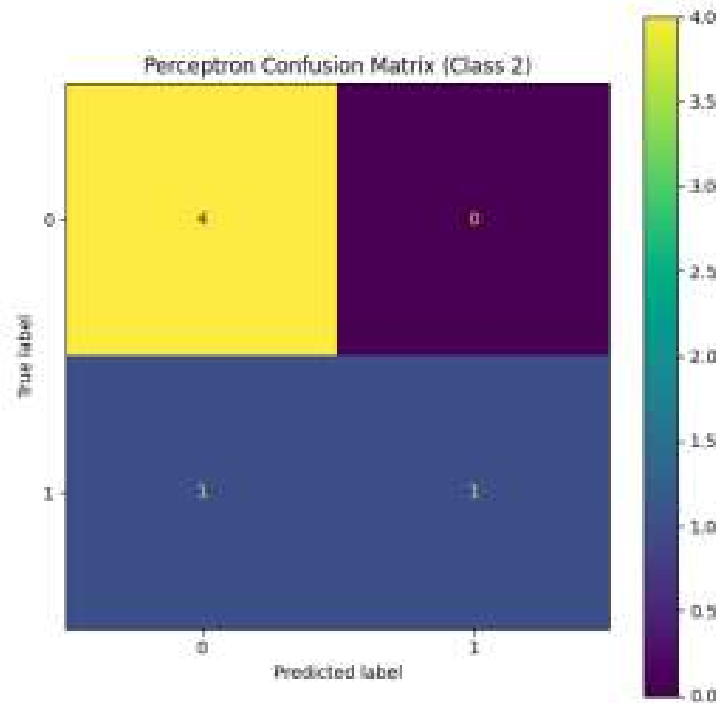
SVM Confusion Matrix (Class 3)



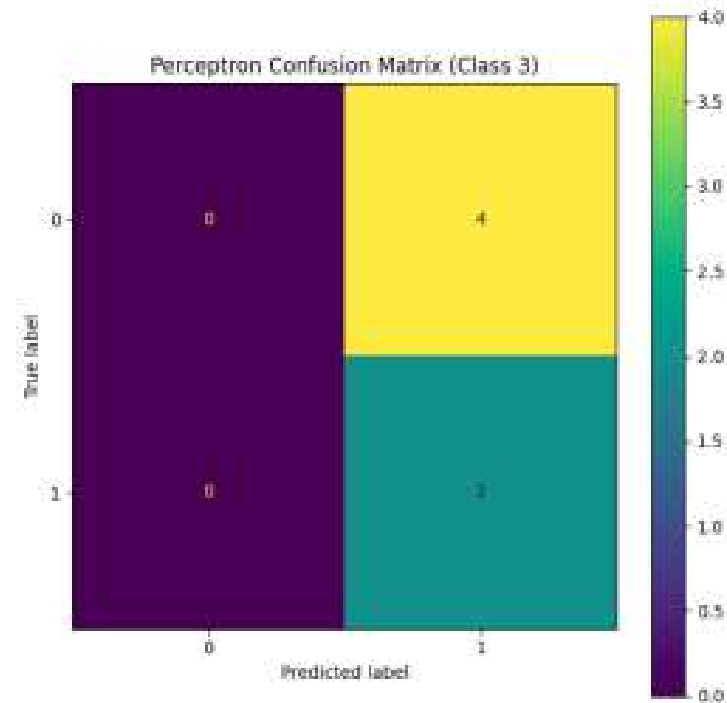
Perceptron Confusion Matrix (Class 1)



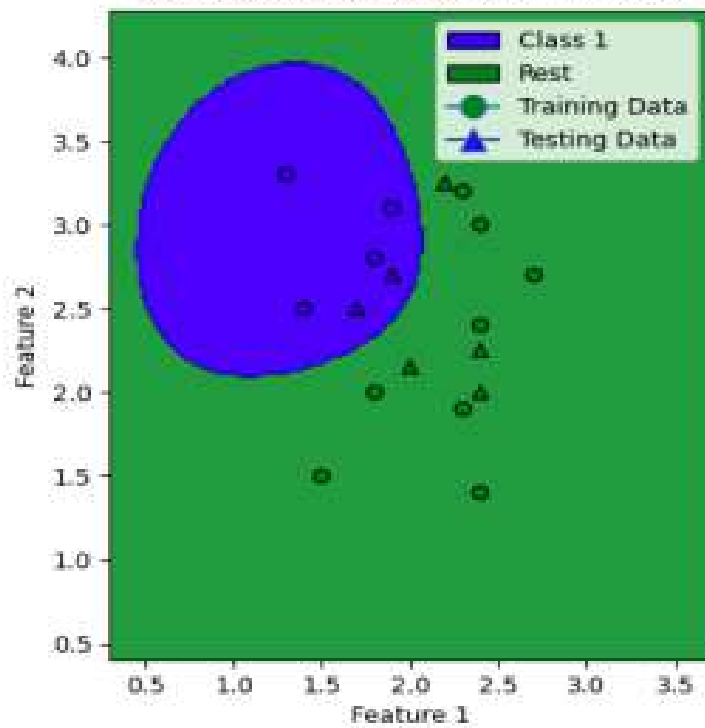
Perceptron Confusion Matrix (Class 2)



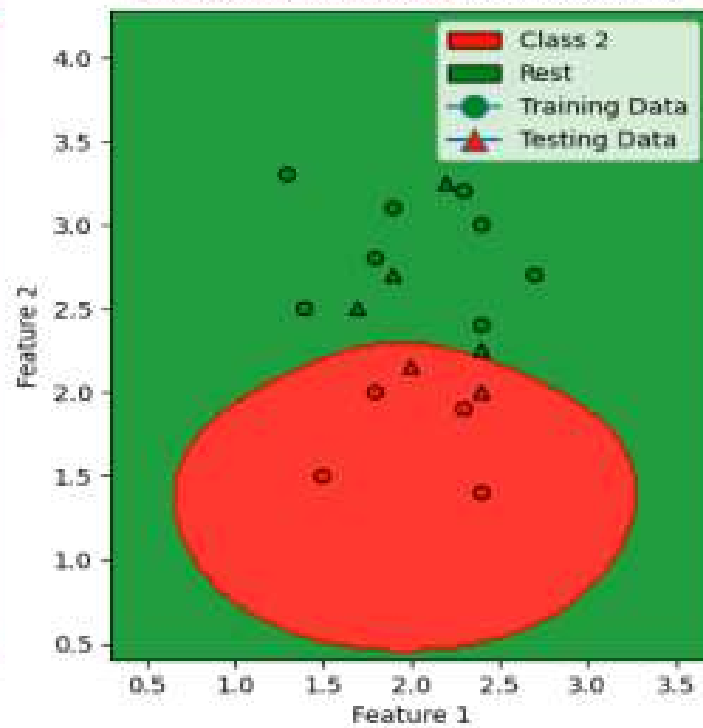
Perceptron Confusion Matrix (Class 3)



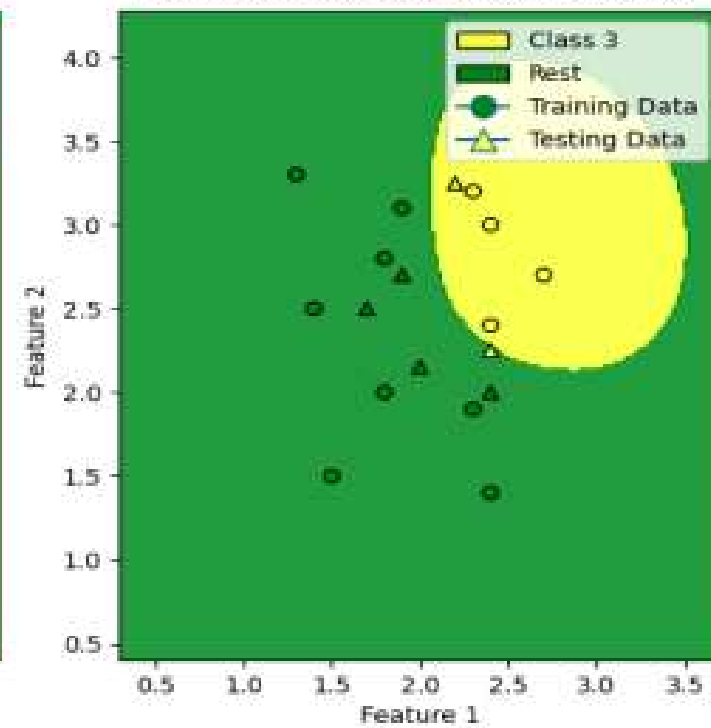
Decision Surfaces for SVM Classifier



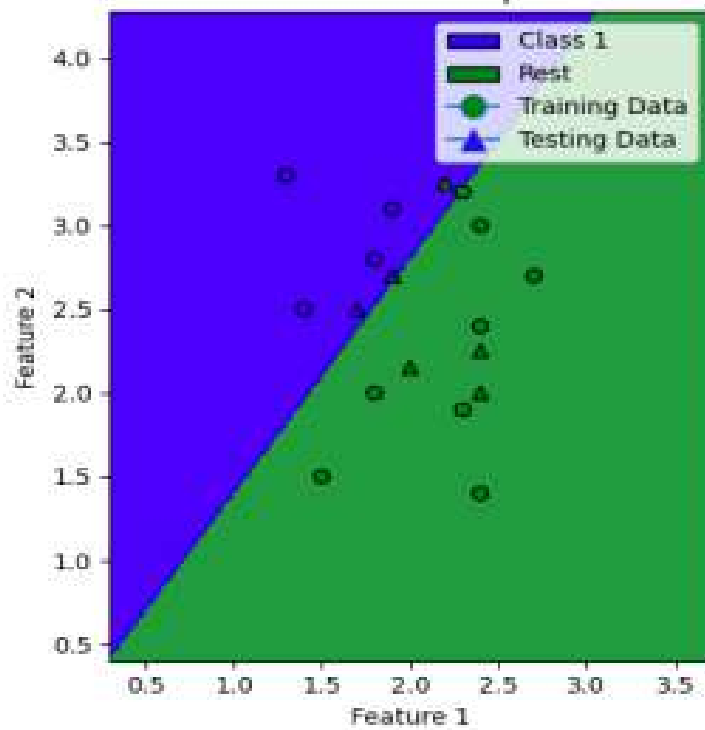
Decision Surfaces for SVM Classifier



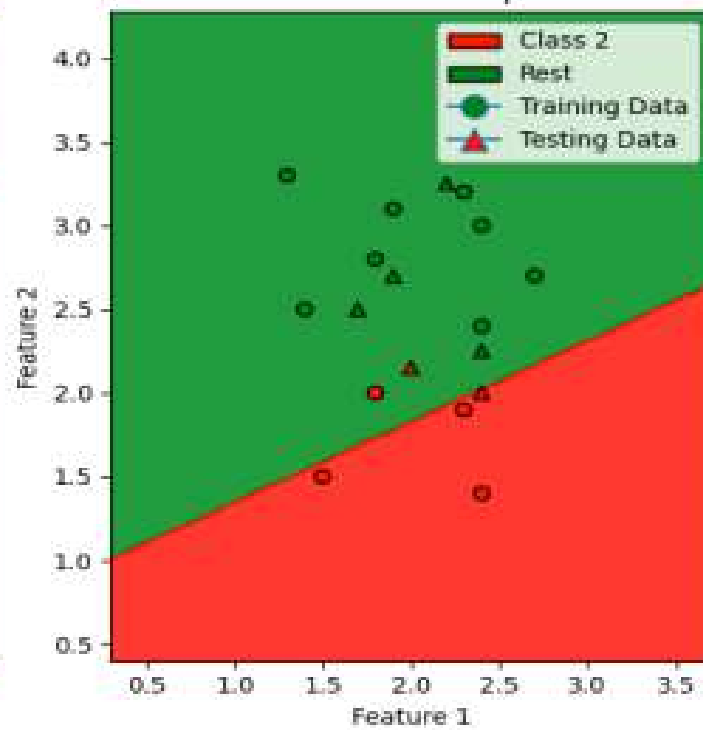
Decision Surfaces for SVM Classifier



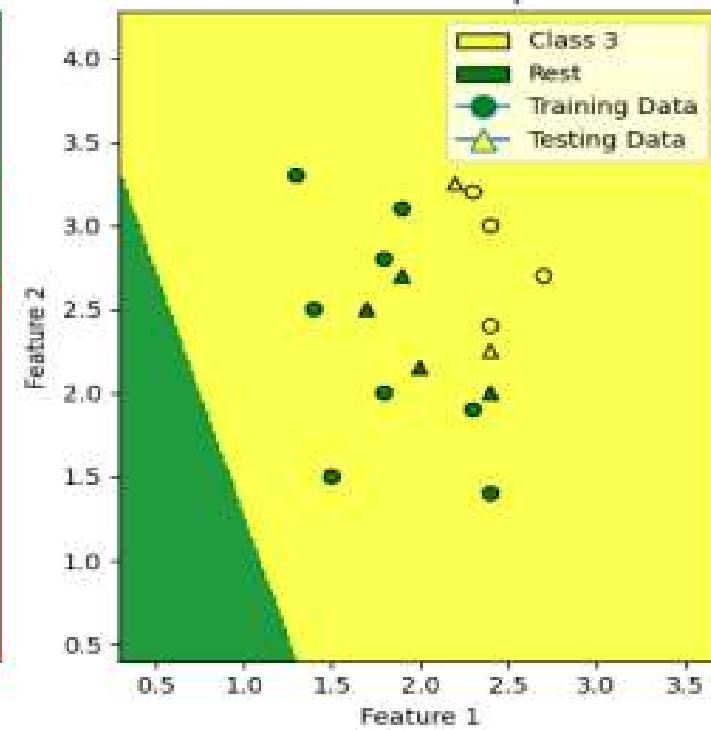
Decision Surfaces for Perceptron Classifier



Decision Surfaces for Perceptron Classifier



Decision Surfaces for Perceptron Classifier

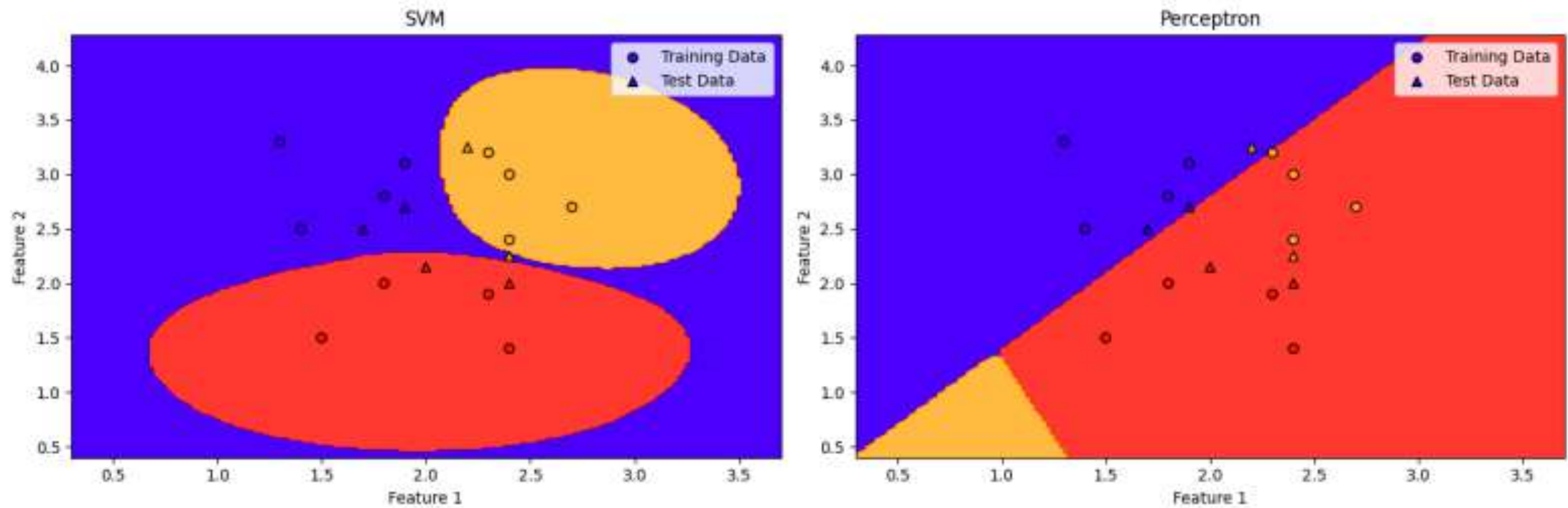


SVM Results: [1.0, 1.0, 0.8333333333333334]

Perceptron Results: [0.8333333333333334, 0.8333333333333334, 0.3333333333333333]

For the default svm

Decision Surfaces for SVM (Default) and Perceptron Classifiers



Analyze performance.

```
# Calculate accuracy score
svm_accuracy = accuracy_score(Y_Test, combined_svm_predictions)
print("Combined SVM Accuracy:", svm_accuracy)

# Calculate accuracy score
perceptron_accuracy = accuracy_score(Y_Test, combined_perceptron_predictions)
print("Combined Perceptron Accuracy:", perceptron_accuracy)
```

```
Combined SVM Accuracy: 0.8333333333333334
Combined Perceptron Accuracy: 0.6666666666666666
```

Compare performance with section (a) solution.

```
print("Combined SVM Accuracy:", svm_accuracy)
print("Combined Perceptron Accuracy:", perceptron_accuracy)

#section (a) accuracy
predictions = classifier.predict(X_Test)
svm_a_accuracy = accuracy_score(Y_Test, predictions)

print("Section (a) Accuracy:", svm_a_accuracy)
```

```
Combined SVM Accuracy: 0.8333333333333334
Combined Perceptron Accuracy: 0.6666666666666666
Section (a) Accuracy: 1.0
```

