Analysis of SVM Performance on 2D Data

Raghda Al taei

2024

1 Introduction

This report presents an analysis of Support Vector Machines (SVM) applied to a 2D dataset. The goal is to classify the data points based on their features and evaluate the model's performance using various kernels and hyperparameters. The data consists of two features and a label indicating the class of each point.

2 Data Visualization

The dataset was visualized in 2D to better understand the distribution of the classes. The scatter plot below displays the data points, color-coded by their respective classes.

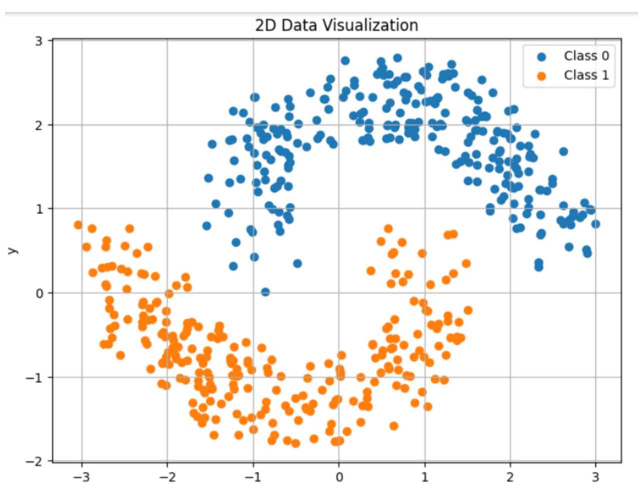

Figure 1: 2D Data Visualization

3 Results and Analysis

3.1 Part 2: Training and Testing

The model was trained and tested with the following results:

- Training set size: 400
- Test set size: 100
- Training Accuracy: 0.97
- Testing Accuracy: 0.97

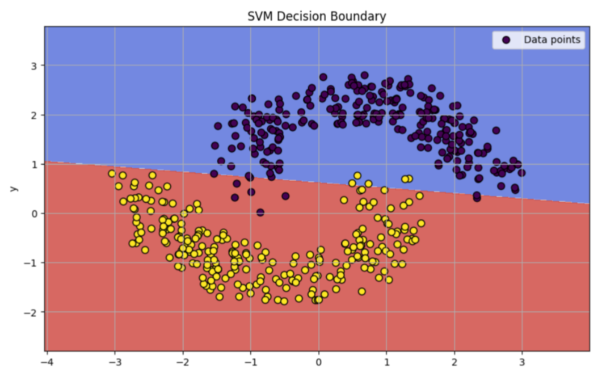

Figure 2: Training Linear SVM

The high accuracy values indicate that the model performs well on both the training and test datasets, suggesting that it is capable of generalizing effectively to unseen data.

3.2 Part 3: SVM Performance with Different C Values

In this part, SVM was trained with different values of the regularization parameter C. The accuracy for all tested C values was consistently 0.98. This indicates that the choice of C had little impact on the model's performance within the tested range, as all configurations led to similarly high accuracy.

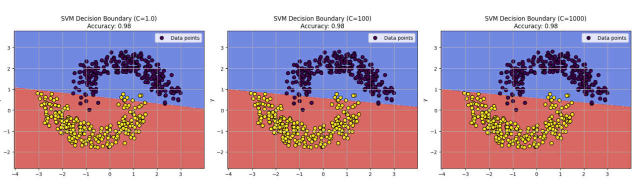

Figure 3: Varying the C Parameter in Linear SVM

Smaller C (C = 1.0): The SVM allows for some misclassifications, resulting in a broader margin and possibly more support vectors.
Intermediate C (C = 100): The SVM tries to find a balance between maximizing the margin and minimizing misclassifications, leading to a tighter fit.
Larger C (C = 1000): The model becomes more sensitive to misclassifications, resulting in a narrower margin and potentially overfitting the training data.

3.3 Part 4: SVM with Polynomial and RBF Kernels

The accuracy of the model using different kernel functions was as follows:

- Polynomial kernel accuracy: 0.97
- RBF kernel accuracy: 1.00

The RBF kernel achieved perfect accuracy, suggesting that it effectively captures the underlying patterns in the data. In contrast, the polynomial kernel performed slightly lower, indicating that it may not be as well-suited for this dataset.

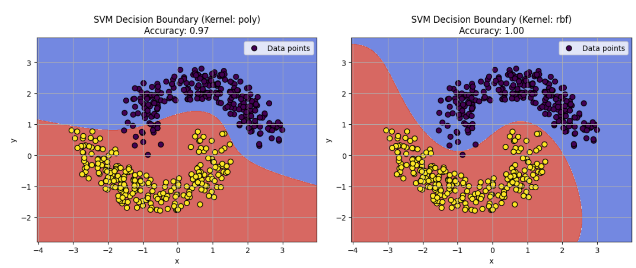

Figure 4: Using Polynomial and RBF Kernels

Polynomial Kernel: This kernel can fit more complex decision boundaries. The behavior will depend on the degree of the polynomial (default is 3), which can lead to more flexibility but may also risk overfitting.

RBF Kernel: The RBF kernel is often a good default choice. It can capture complex relationships in the data. You may need to adjust the parameter for optimal performance.

3.4 Part 5: Hyperparameter Optimization

The best parameters found for the SVM model were:

```
Best parameters found:  {'C': 1, 'gamma': 1}
```

The classification report showed excellent performance with:

- Precision: 1.00

- Recall: 1.00
- F1-Score: 1.00
- Support: 75

These metrics indicate that the model has perfect precision and recall, effectively classifying all instances in the test set without any false positives or false negatives.

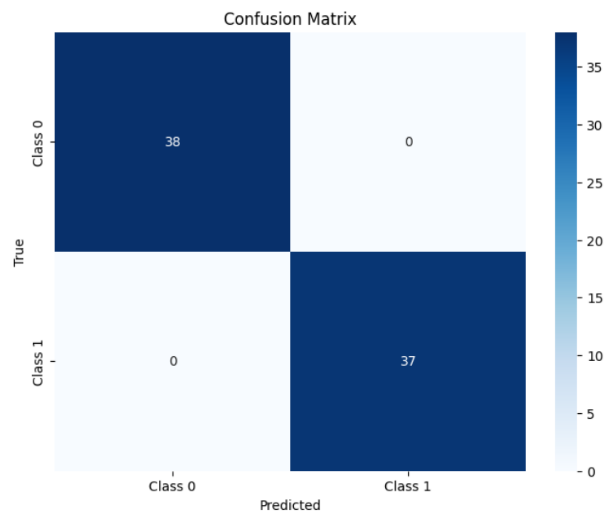

Figure 5: Hyperparameter Optimization confusion matrices

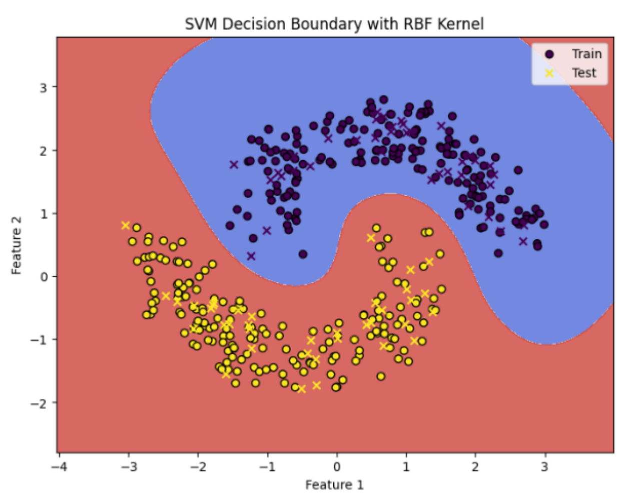

Figure 6: Hyperparameter Optimization decision boundaries

3.5 Part 6: ROC Curve Analysis

The ROC curve analysis yielded the following results:

- False Positive Rate (FPR): [0.0, 0.0, 0.0, 1.0]
- True Positive Rate (TPR): [0.0, 0.027, 1.0, 1.0]
- Area under the ROC curve (AUC): 1.0

The AUC of 1.0 indicates perfect classification capability of the model across various thresholds. The TPR increases significantly while maintaining a low FPR, demonstrating the model's effectiveness in distinguishing between the classes.

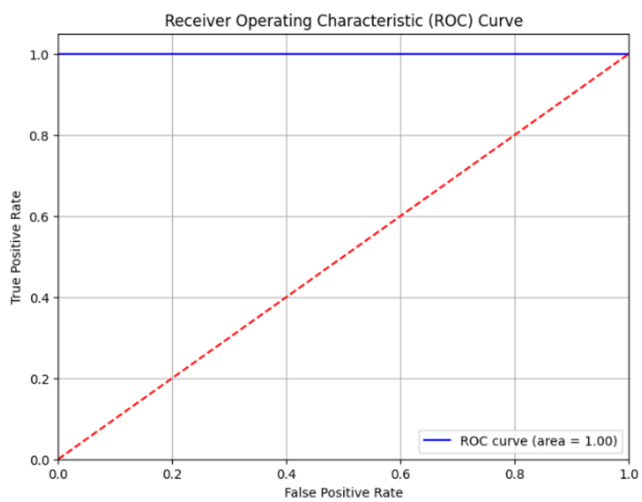

Figure 7: ROC Curve

4 Conclusion

The SVM model exhibited strong performance across all phases of the analysis. With high accuracy, excellent precision and recall, and a perfect AUC score, the model is well-suited for classifying the given 2D dataset. Future work could involve testing on more complex datasets or exploring different hyperparameter tuning strategies to further enhance performance.