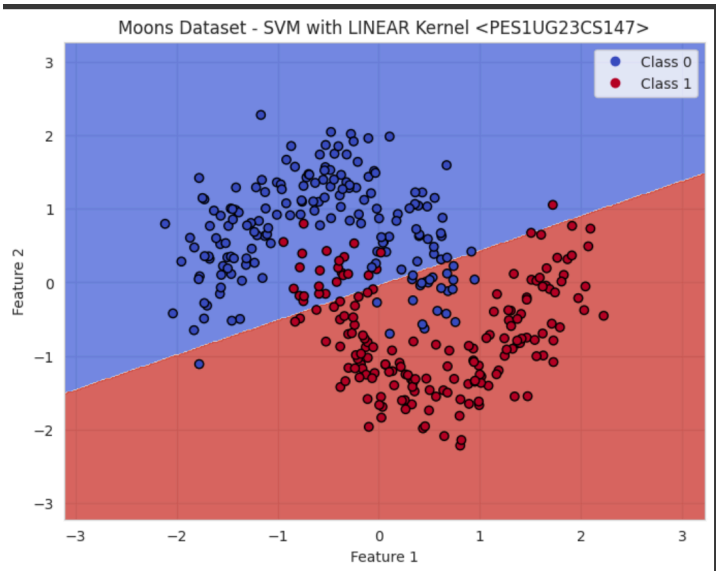
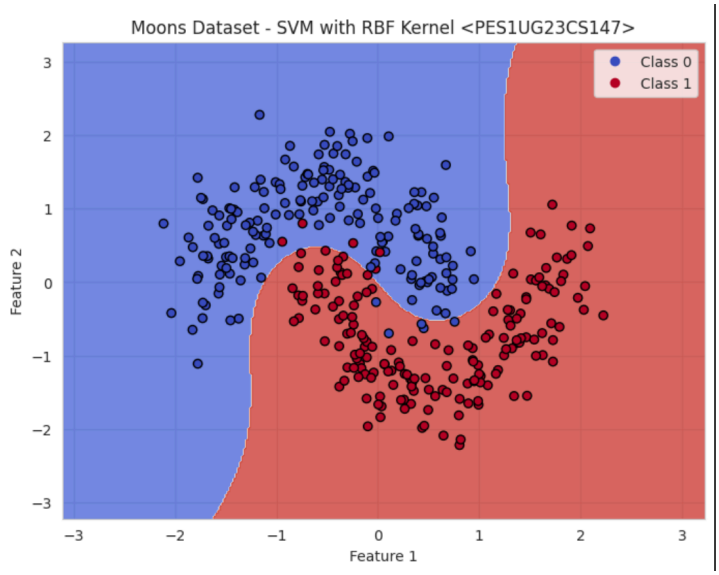
ML Week 10
SVM

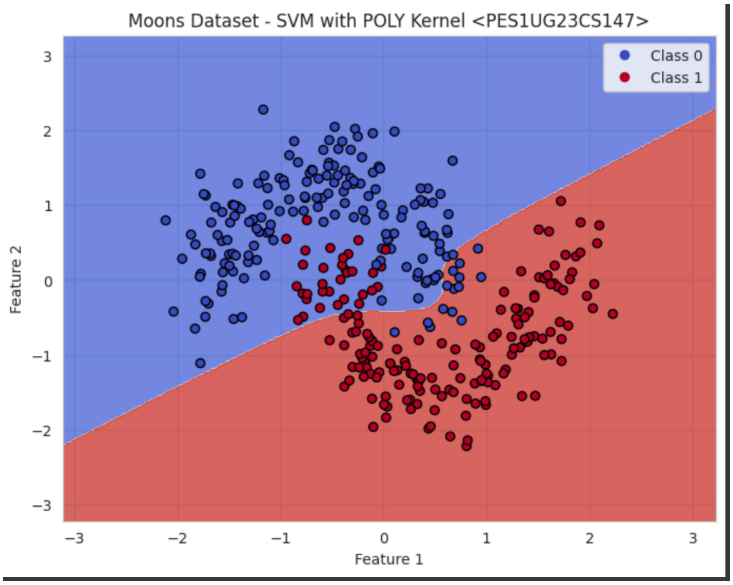
Name: Cheruku Manas Ram
SRN: PES2UG23CS147
Section: 5 ‘C’
Date: 10/10/2025

Moons Dataset

```
SVM with LINEAR Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

           0       0.85      0.89      0.87        75
           1       0.89      0.84      0.86        75

    accuracy                           0.87       150
   macro avg       0.87      0.87      0.87       150
weighted avg       0.87      0.87      0.87       150

----------------------------------------

SVM with RBF Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

           0       0.95      1.00      0.97        75
           1       1.00      0.95      0.97        75

    accuracy                           0.97       150
   macro avg       0.97      0.97      0.97       150
weighted avg       0.97      0.97      0.97       150

----------------------------------------

SVM with POLY Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

           0       0.85      0.95      0.89        75
           1       0.94      0.83      0.88        75

    accuracy                           0.89       150
   macro avg       0.89      0.89      0.89       150
weighted avg       0.89      0.89      0.89       150

----------------------------------------
```


Moons Dataset - SVM with LINEAR Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2


Moons Dataset - SVM with RBF Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2

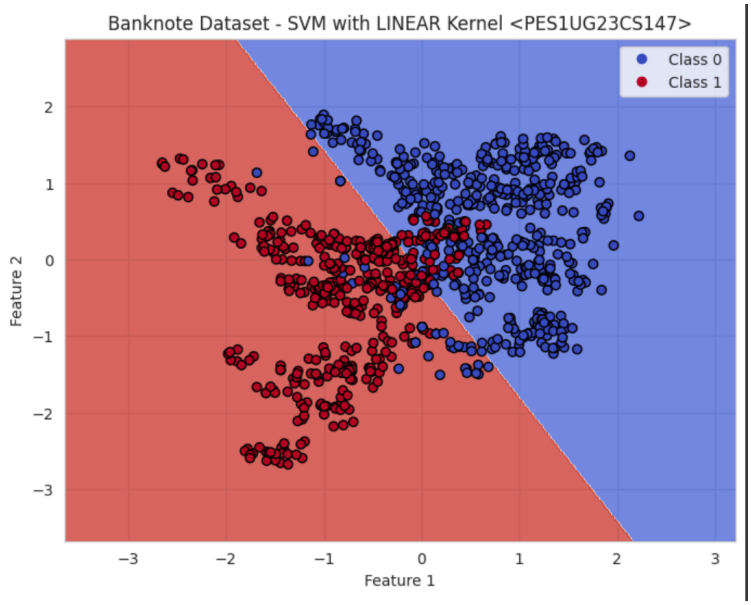

Moons Dataset - SVM with POLY Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2

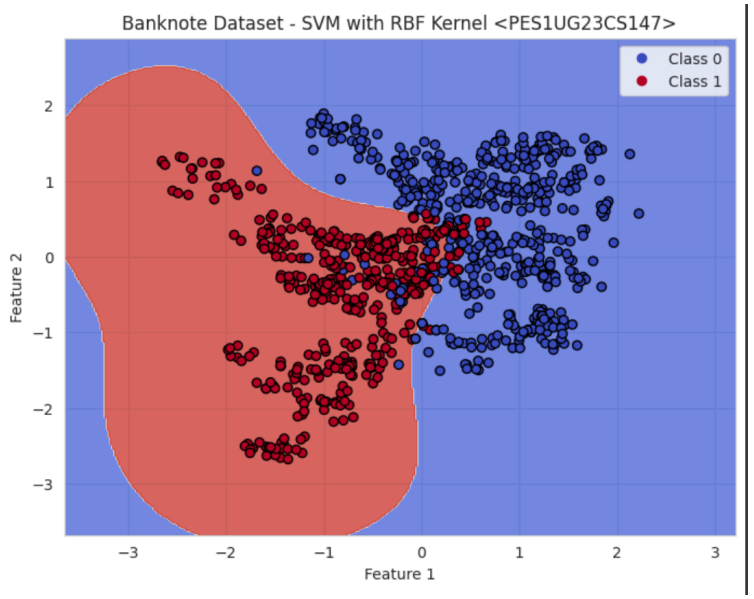
1. The Linear Kernel performs reasonably well, achieving an accuracy of 0.87. However, looking at the visualisation, the linear decision boundary which is a straight line, is not able to perfectly separate the two interlocking moon shapes. This is shown in the classification report, where the precision and recall for both classes are around 0.85-0.89, indicating some misclassifications.

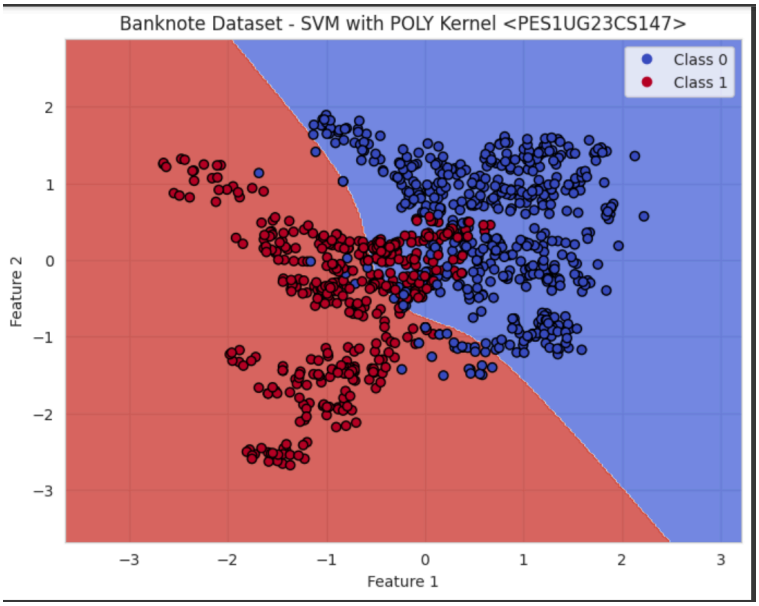
2. Both the RBF and Polynomial kernels create non-linear decision boundaries that are much better suited than the linear kernel. Comparing their decision boundaries, the RBF kernel seems to capture the shape of the data more naturally. Its boundary follows the curves of the two moons, resulting in an almost perfect separation of the classes. The Polynomial kernel also creates a curved boundary, but it appears less smooth and doesnot conform as closely to the data's structure compared to the RBF kernel in this specific case. The classification reports also support this, with the RBF kernel achieving a higher accuracy (0.97) compared to the Polynomial kernel (0.89).

Banknotes Dataset

```
SVM with LINEAR Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

      Forged       0.90      0.88      0.89       229
     Genuine       0.86      0.88      0.87       183

    accuracy                           0.88       412
   macro avg       0.88      0.88      0.88       412
weighted avg       0.88      0.88      0.88       412

----------------------------------------

SVM with RBF Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

      Forged       0.96      0.91      0.94       229
     Genuine       0.90      0.96      0.93       183

    accuracy                           0.93       412
   macro avg       0.93      0.93      0.93       412
weighted avg       0.93      0.93      0.93       412

----------------------------------------

SVM with POLY Kernel <PES1UG23CS147>
              precision    recall  f1-score   support

      Forged       0.82      0.91      0.87       229
     Genuine       0.87      0.75      0.81       183

    accuracy                           0.84       412
   macro avg       0.85      0.83      0.84       412
weighted avg       0.85      0.84      0.84       412

----------------------------------------
```


Banknote Dataset - SVM with LINEAR Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2

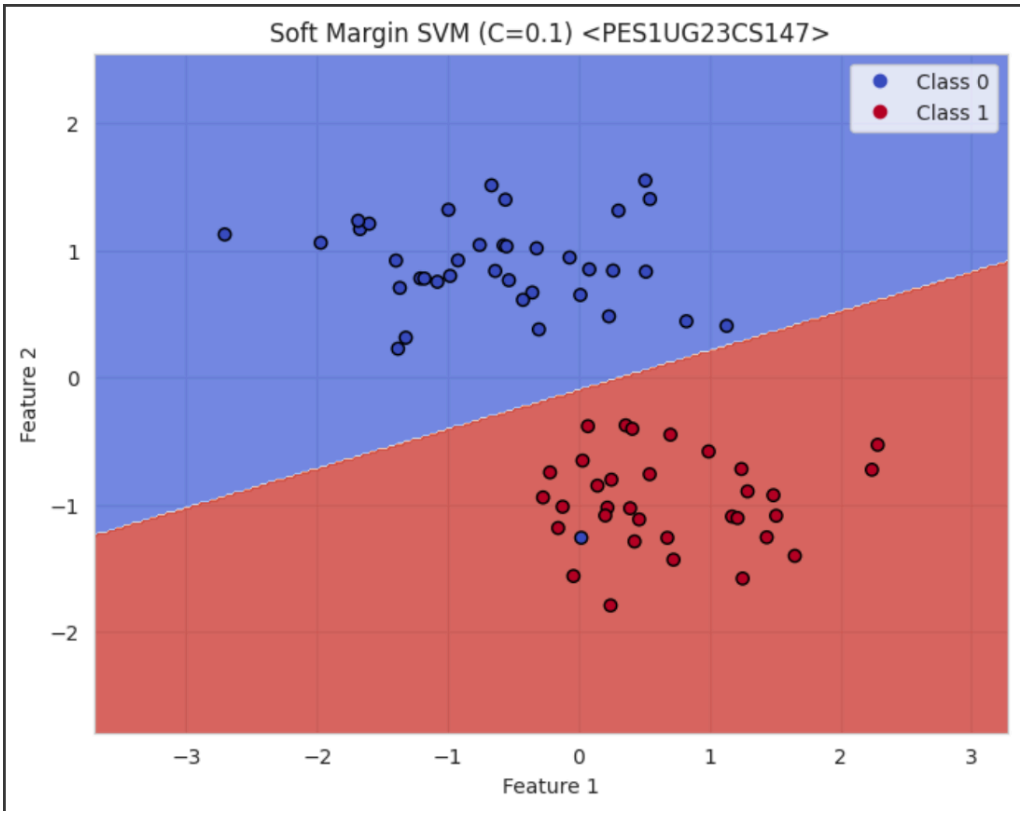

Banknote Dataset - SVM with RBF Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2

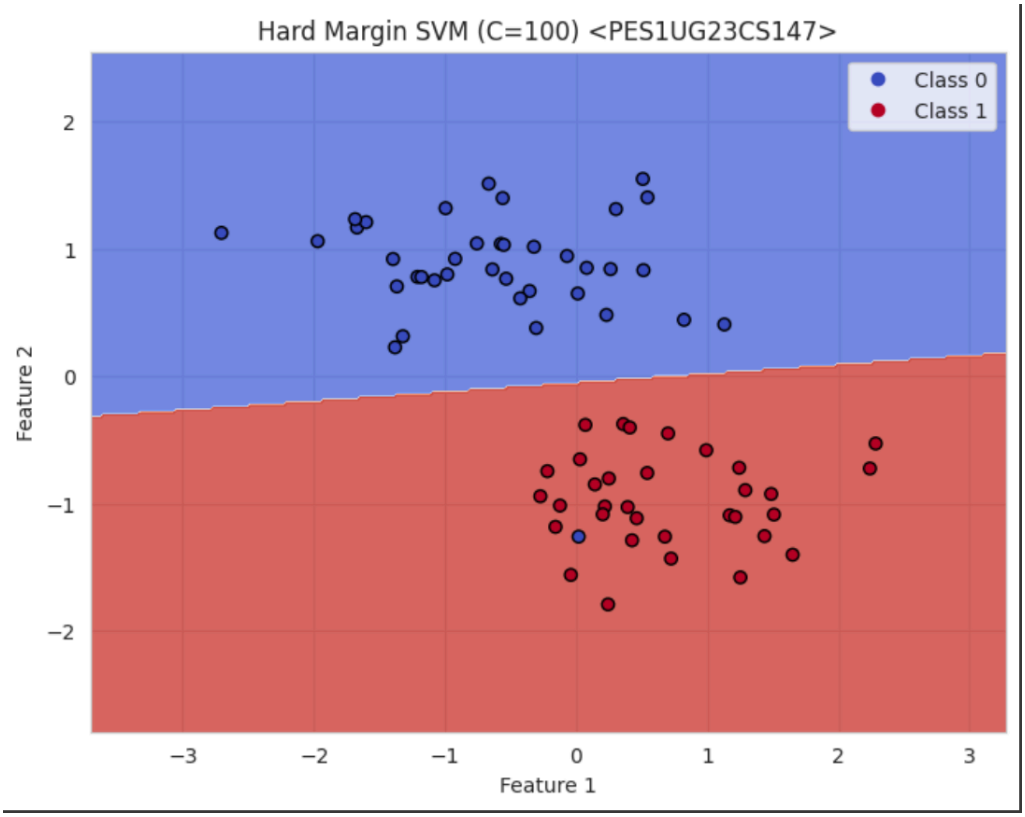

Banknote Dataset - SVM with POLY Kernel <PES1UG23CS147>
Class 0
Class 1
Feature 1
Feature 2

1. The RBF kernel is the most effective for this dataset. It achieved the highest overall accuracy (0.93) and higher precision and recall values for both 'Forged' and 'Genuine' classes compared to the Linear and Polynomial kernels. The visualization also shows the RBF kernel creating a decision boundary that seems to separate the two classes quite well.

2.
- We can observe that the data points in moons is less complex than banknotes. Hence, it might be easier to the polynomial model to converge better in moons, unlike in banknotes where the seperation of the datapoints is more complex, with a lot more points overlapping on each other, making the accuracy of polynomial in banknotes lower.
- The polynomial kernel creates decision boundaries based on polynomial combinations of the features. In the banknotes dataset, we use only two features ('variance' and 'skewness'), the polynomial combinations might not be sufficient to distinguish between forged and genuine banknotes compared to the RBF kernel's approach of mapping data to an infinite-dimensional space.

Hard vs Soft Margin





1. Looking at the graph, we can observe that the hard margin SVM (C=100) produces a wider margin even though in theory, the soft margin SVM should produce a wider margin.

2. The SVM with a soft margin allows these mistakes because its goal is not to perfectly classify every single training point, instead to find a decision boundary that generalizes well to unseen data. By allowing some misclassifications (controlled by the C parameter), the model can find a wider margin which is less sensitive to outliers. The primary goal is to maximize the margin while keeping the number of margin violations below a certain threshold determined by C.

3. The "Hard Margin" model (C=100) is more likely to be overfitting to the training data. With a very large C value, the model is heavily penalised for any misclassification. It tries to create a very narrow margin that perfectly separates all or almost all training points, including potential outliers. This can lead to a complex decision boundary that captures the noise in the training data and may not perform well on new, unseen data.

4. I would generally trust the "Soft Margin" model (C=0.1) more to classify a new, unseen data point correctly. This is the soft margin model is less likely to have overfit to the training data. Its wider margin means better generalisation.
In a real-world scenario, we would generally prefer to start with a low value of C. A low C value creates a soft margin, which is more tolerant of noise. This approach helps to prevent overfitting and is more likely to result in a model that performs well on real-world, imperfect data.