MACHINE LEARNING LAB

SUPPORT VECTOR MACHINE

WEEK-10

NAME: B TEJA DEEP SAI KRISHNA

SRN: PES2UG23CS135

SECTION: 5C

Moons Dataset Questions:

1. Inferences about the Linear Kernel's performance.
    - Linear kernel performs poorly on the Moons dataset (accuracy ~87%)
    - Cannot capture the non-linear relationship between the two interlocking half-moons
    - Creates a straight-line decision boundary that cannot properly separate the curved moon shapes
    - Struggles with the non-linearly separable nature of this synthetic dataset
    - Demonstrates the limitation of linear kernels when data has complex, curved patterns
2. Comparison between RBF and Polynomial kernel decision boundaries.
    - RBF kernel (accuracy ~97%) creates smooth, curved boundaries that naturally follow the moon shapes
    - Polynomial kernel (accuracy ~89%) creates more angular, polynomial-shaped boundaries
    - RBF captures the data shape more naturally because it can create smooth curves and circles
    - RBF is more flexible in adapting to the specific curvature of the moons
    - Polynomial kernel tends to create more rigid, mathematical curve patterns that don't fit as well

Banknote Dataset Questions:

1. Which kernel was most effective for this dataset?
    - RBF kernel was most effective (accuracy ~93%)
    - Linear kernel performed well (accuracy ~88%) but not as good as RBF
    - Polynomial kernel underperformed (accuracy ~84%)
    - RBF kernel was able to capture the subtle non-linear patterns in the banknote features
2. Why might the Polynomial kernel have underperformed here?
    - Banknote data is more linearly separable than the Moons dataset
    - Polynomial kernel may be overfitting to the training data with its complex polynomial curves
    - The data doesn't have the same curved structure as the Moons dataset
    - Polynomial kernel's complexity is unnecessary for this relatively simple separation task
    - RBF kernel's smooth boundaries work better for the banknote feature space

Hard vs. Soft Margin Questions:

1. Which margin (soft or hard) is wider?
    - Soft margin (C=0.1) produces a wider margin
    - Hard margin (C=100) produces a narrower margin
    - Soft margin allows more space between the decision boundary and support vectors
    - Hard margin tries to maximize the margin while still classifying all points correctly
2. Why does the soft margin model allow "mistakes"?
    - Primary goal is better generalization, not perfect training accuracy
    - Trades some training accuracy for better performance on new data
    - More robust to outliers and noise in the dataset
    - Prevents overfitting by not forcing perfect classification of every training point
    - Creates a more stable decision boundary that's less sensitive to individual data points
3. Which model is more likely to be overfitting and why?
    - Hard margin model (C=100) is more likely to overfit

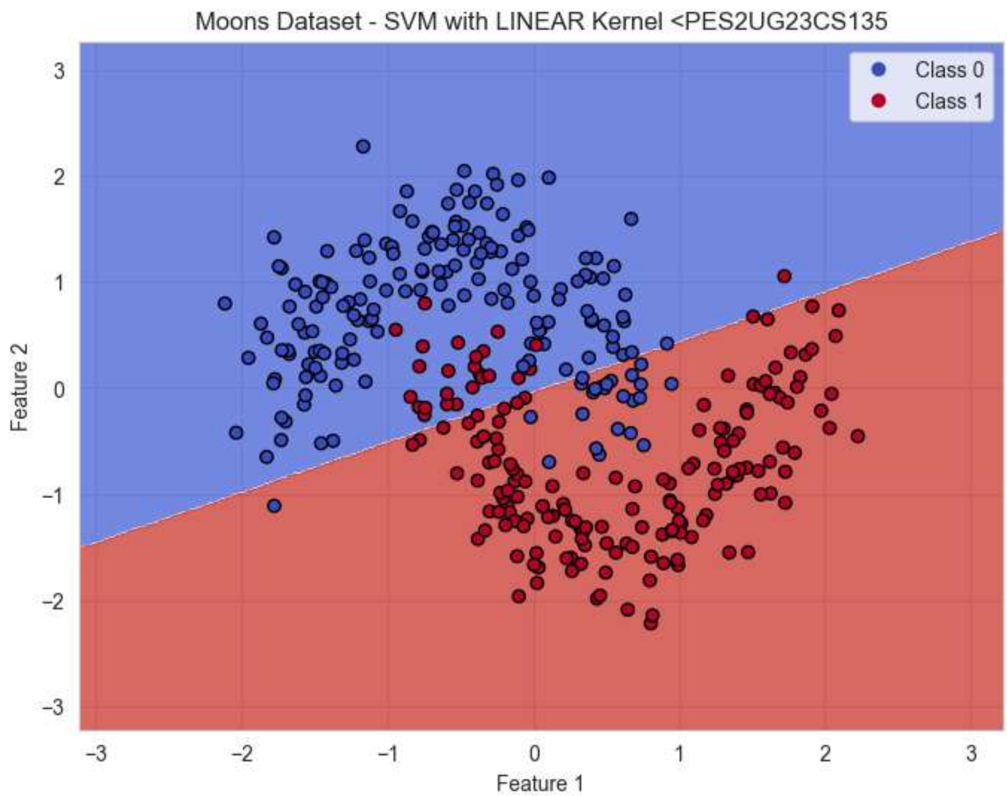
- Tries to classify every training point perfectly, including outliers
- Creates a very specific decision boundary that may not generalize well
- Sensitive to noise and outliers in the training data
- Less regularization means it can memorize the training data patterns

SCREENSHOTS:

MOONS DATASET:

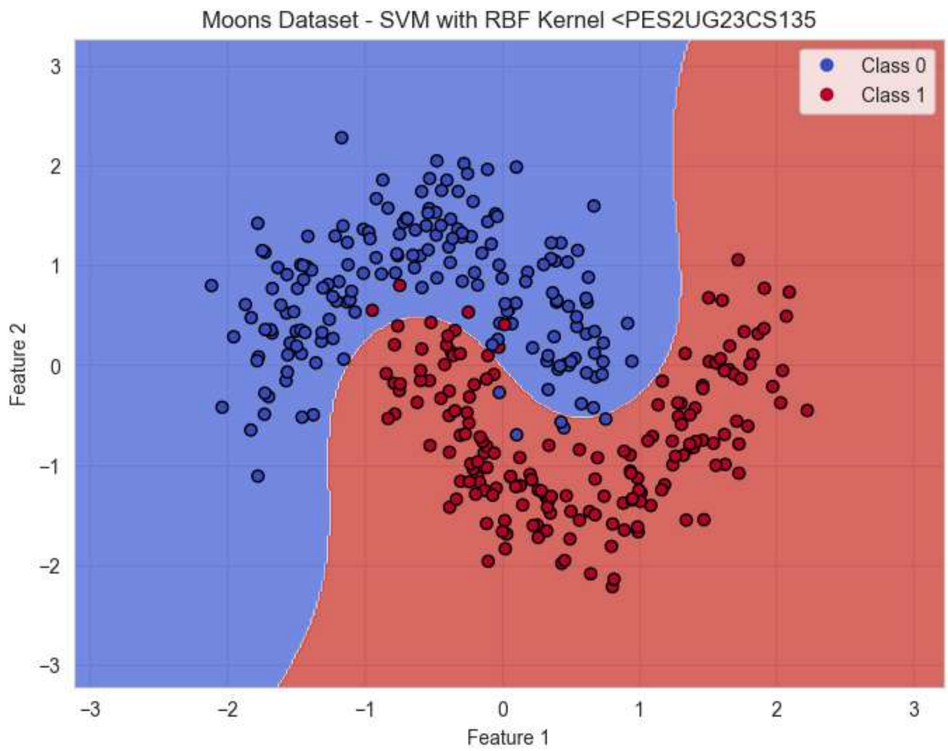
LINEAR KERNEL:

```
SVM with LINEAR Kernel <PES2UG23CS135>
              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

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



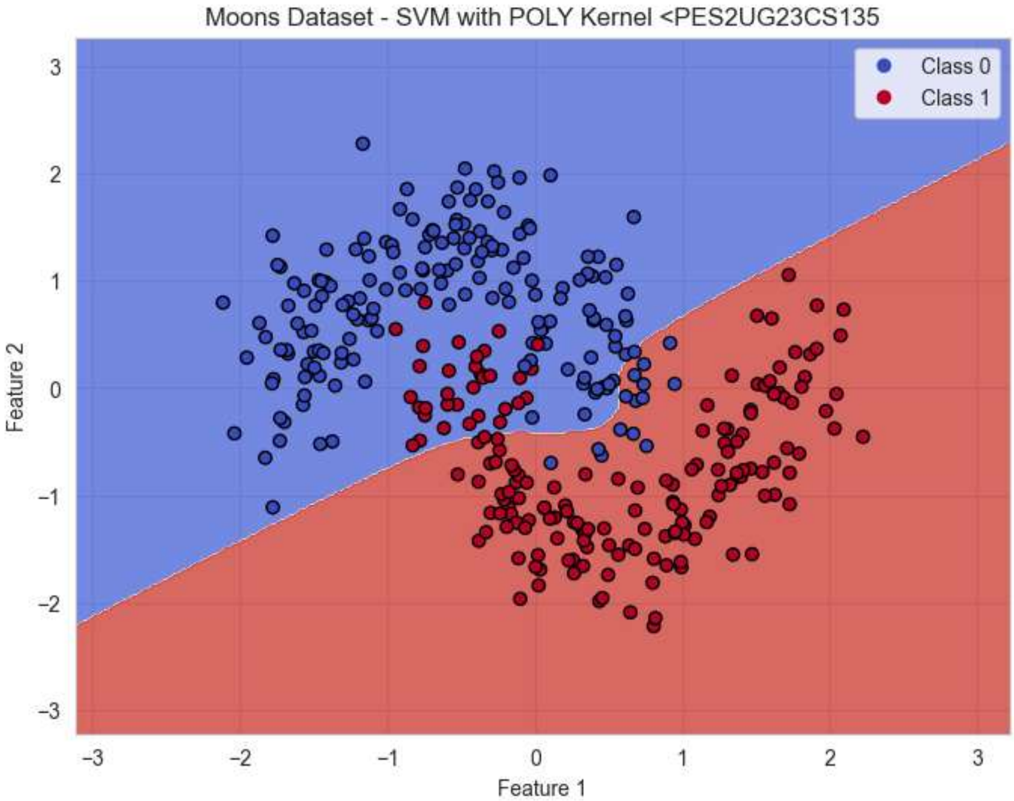
RBF KERNEL:

```
SVM with RBF Kernel <PES2UG23CS135>
              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


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



POLYNOMIAL KERNEL:

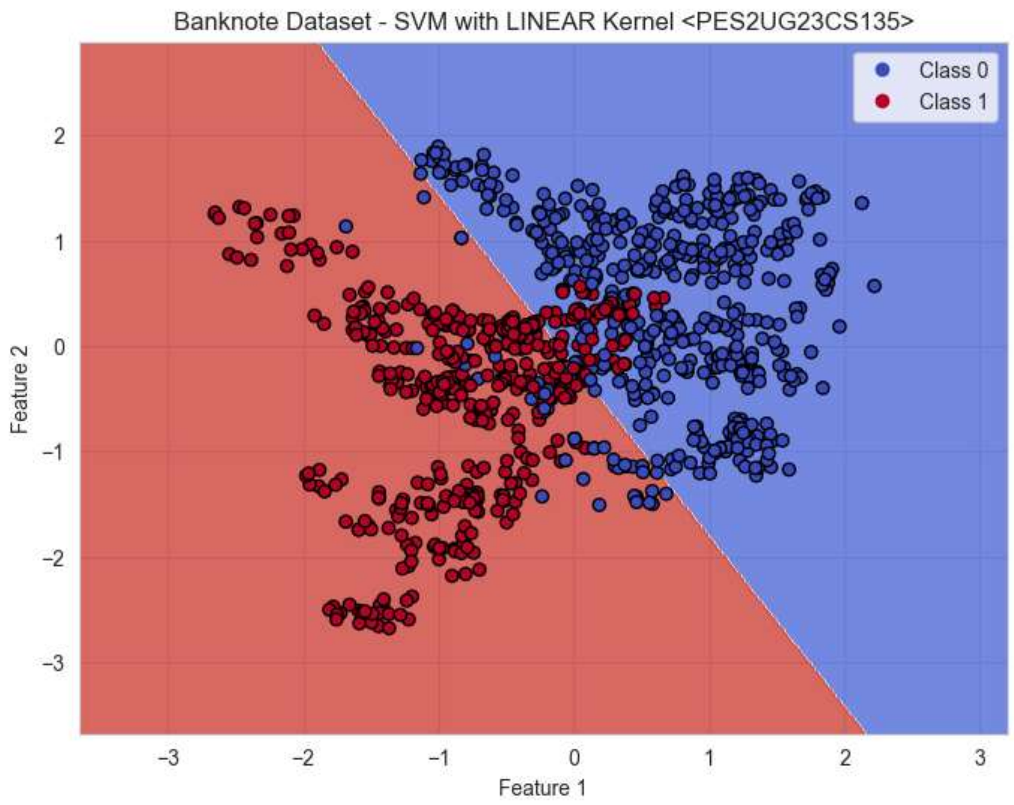
```
SVM with POLY Kernel <PES2UG23CS135>
              precision    recall  f1-score   support

           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

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



Banknote Dataset:

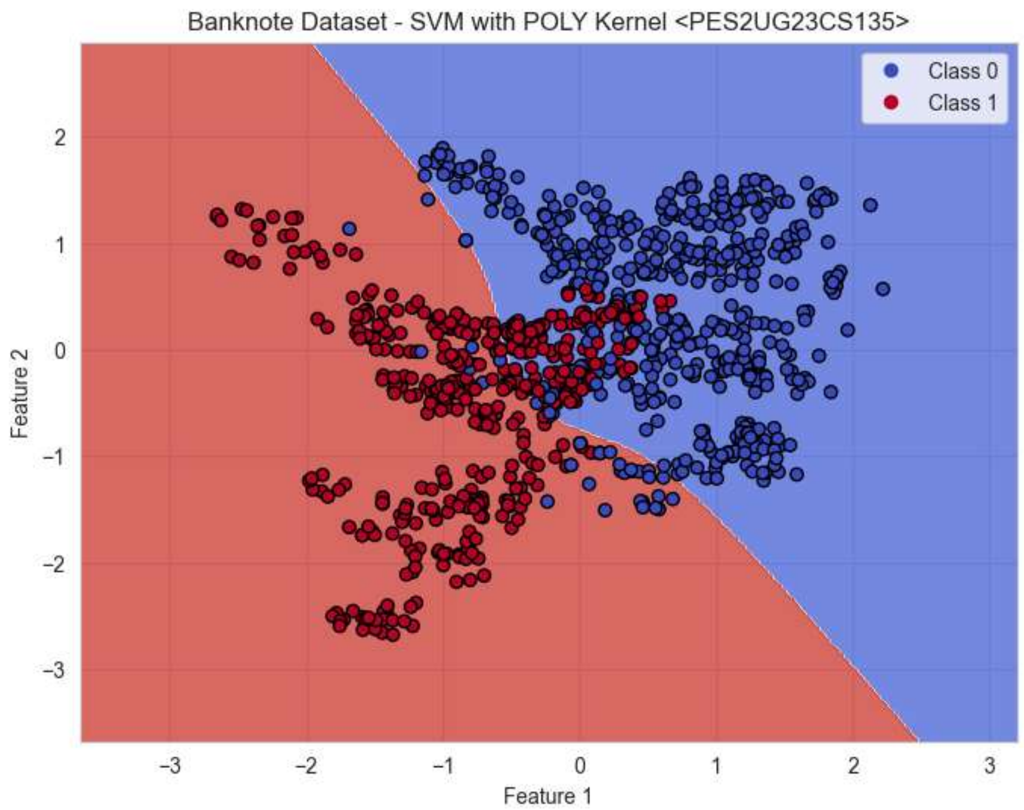
Linear Kernel:

```
SVM with LINEAR Kernel <PES2UG23CS135>
              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


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



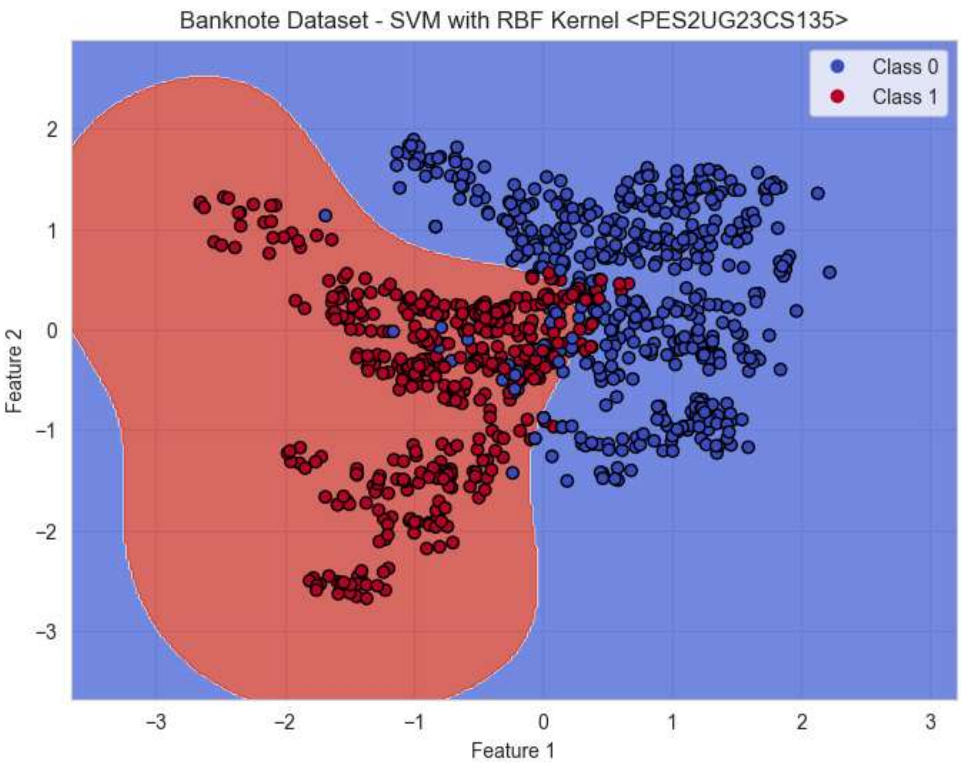
Polynomial kernel:

```
SVM with POLY Kernel <PES2UG23CS135>
              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


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

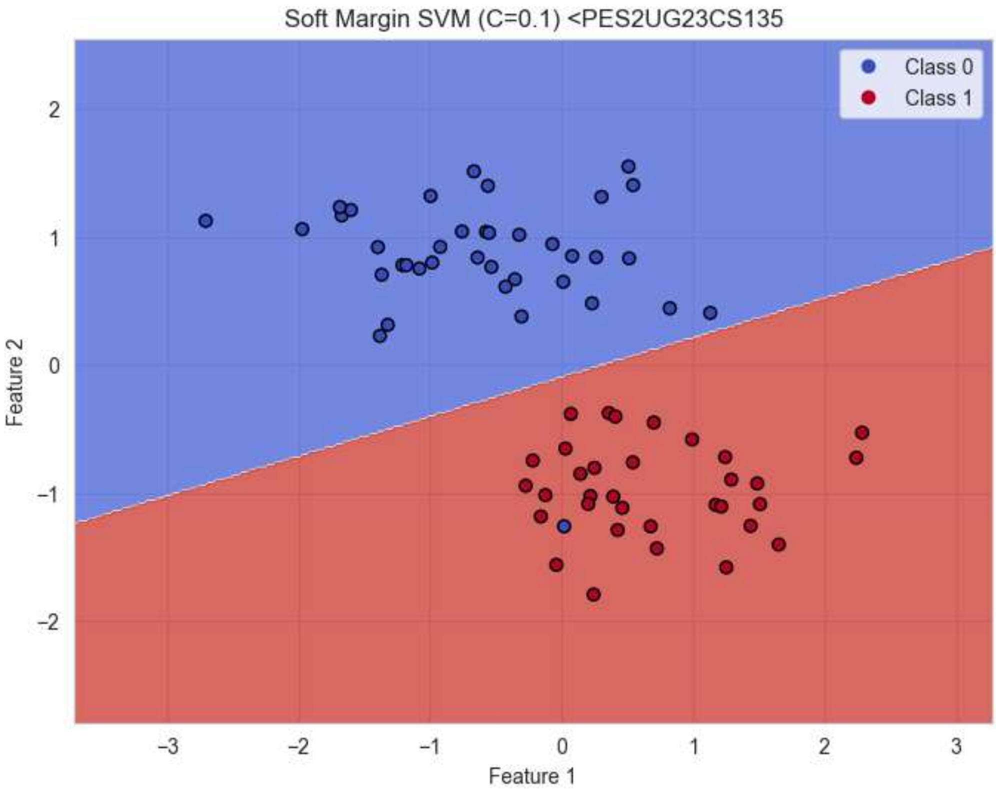


RBF Kernel:

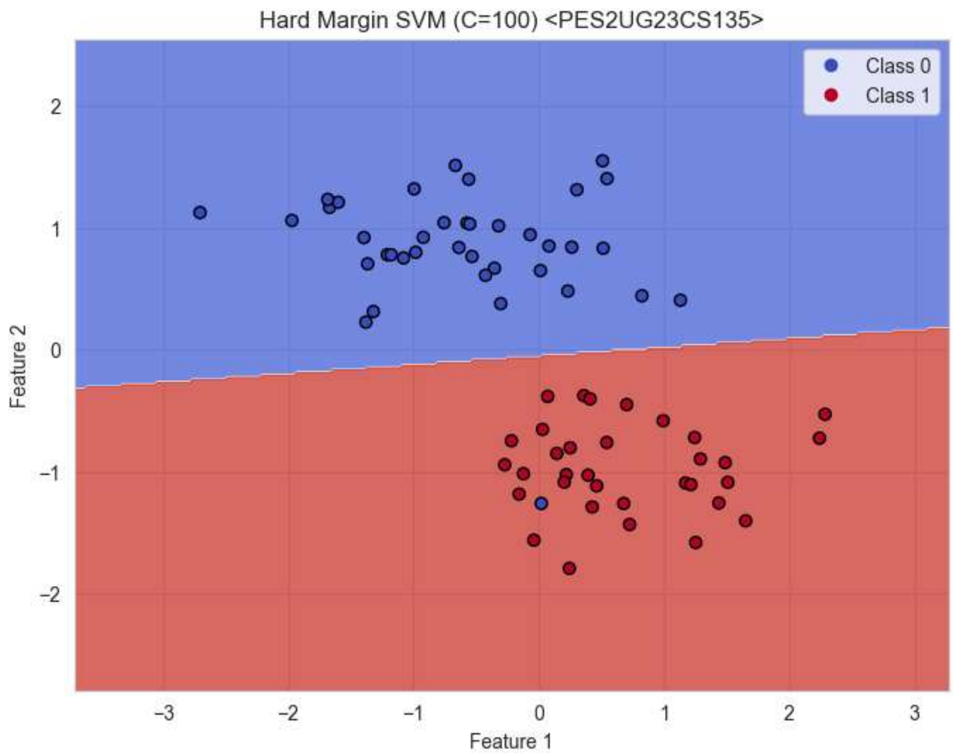
```
SVM with RBF Kernel <PES2UG23CS135>
              precision    recall  f1-score   support

      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


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



SOFT-MARGIN:



HARD-MARGIN:


Hard Margin SVM (C=100) <PES2UG23CS135>
Class 0
Class 1
Feature 2
2
1
0
−1
−2
−3
−2
−1
0
1
2
3
Feature 1