Machine Learning Lab - Week 10: Support Vector Machines (SVM)

Name: Mitha M K

SRN: PES2UG23CS339

Section: F

Course: Machine Learning Laboratory

Experiment Title: Support Vector Machine Classifier

1. Objective

The goal of this lab was to understand how **Support Vector Machines (SVMs)** can classify data by finding an optimal separating boundary.
We explored three different kernels — **Linear**, **RBF**, and **Polynomial** — to observe how each performs on different types of datasets.
Additionally, we compared **Soft Margin** and **Hard Margin** SVMs to see how the margin parameter CCC affects performance and generalization.

---

2. Concept Overview

Support Vector Machine (SVM): A supervised learning algorithm that finds the hyperplane which best separates classes with the largest possible margin.

Kernel Trick: Transforms input data into a higher-dimensional space so that even non-linear data can become separable.

Linear Kernel: Creates a straight-line decision boundary; best for linearly separable data.

RBF Kernel: Produces non-linear, flexible boundaries that can handle complex datasets.

Polynomial Kernel: Fits curved boundaries based on polynomial degrees.

Hard vs. Soft Margin:

Hard Margin (High C) – tries to classify all points correctly but may overfit.

Soft Margin (Low C) – allows some misclassifications but generalizes better.

3. Datasets Used

(a) Moons Dataset

This is a synthetic dataset generated using `make_moons()` from Scikit-learn.
It contains 500 samples with a small amount of noise, shaped like two interlocking half-moons.
It's an ideal dataset for testing how well non-linear kernels like RBF and Polynomial perform.

(b) Banknote Authentication Dataset

This is a real-world dataset from the **UCI Machine Learning Repository**, used to detect whether a banknote is genuine or forged.
The original data includes features like **variance**, **skewness**, **curtosis**, and **entropy**.
For visual clarity, only **variance** and **skewness** were used to plot the decision boundaries.

> *Note:* During my lab session, I wasn't able to load the dataset directly from the UCI URL due to network restrictions in the Jupyter environment.
> To continue with the experiment, I used a local fallback approach and verified the rest of the SVM implementation successfully.

---

4. Implementation Summary

Moons Dataset

Three SVM classifiers were trained using:

Linear Kernel

RBF (Radial Basis Function) Kernel

Polynomial Kernel

Each model was trained, tested, and evaluated using a classification report and a decision boundary visualization.

Kernel	Accuracy	Observation
Linear	Moderate	Couldn't properly capture the curved moon-shaped boundary.
RBF	High	Captured non-linear separation very effectively.
Polynomial	Average	Worked okay, but slightly sensitive to noise and degree settings.

The **RBF kernel** clearly performed the best, producing smooth and natural boundaries between the two classes.

Banknote Dataset

For the Banknote dataset, three SVMs were also trained with the same kernels.

Kernel	Accuracy	Observation
Linear	Very high	The classes were almost linearly separable.
RBF	Slightly better	Handled edge cases more smoothly.
Polynomial	Lower	Overfitted slightly; unnecessary complexity.

The **Linear kernel** was sufficient for this dataset because the data itself was mostly separable using a straight hyperplane.

Soft vs. Hard Margin Analysis

To visualize the concept of margins, I generated a small linear dataset with some noise and trained two SVMs:

Soft Margin: C = 0.1

Hard Margin: C = 100

Model	Margin Width	Misclassifications	Overfitting Risk
Soft Margin	Wider	Allows few mistakes	Low
Hard Margin	Narrow	No mistakes	High

The soft margin created a smoother, wider boundary that tolerated outliers, while the hard margin fit the training data too tightly.

5. Answers to Analysis Questions

Moons Dataset

Q1. Which kernel was most effective for this dataset?
A1. The **RBF kernel** performed best because it adapts well to non-linear boundaries like the interlocking moons.

Q2. Why might the Polynomial kernel have underperformed here?
A2. The polynomial kernel tends to overfit or underfit depending on its degree. It struggles when the data has irregular non-linear structures that aren't polynomial in nature.

Banknote Dataset

Q3. Which kernel appeared most effective for this dataset?
A3. The **Linear kernel** performed very well because the classes are almost linearly separable in the variance-skewness space.

Q4. Why did the Polynomial kernel perform worse here?
A4. Polynomial kernels introduce extra curvature, which isn't necessary for this dataset, leading to a slightly less general model.

Hard vs. Soft Margin

Q5. Which margin (soft or hard) is wider?
A5. The **soft margin (C=0.1)** is wider because it allows some misclassifications to achieve better generalization.

Q6. Why does the soft margin allow mistakes?
A6. Because its goal is to maximize the overall margin width, even if that means a few points fall on the wrong side of the decision boundary.

Q7. Which model is more likely to overfit?
A7. The **hard margin (C=100)** model, since it tries to perfectly classify all training points, including noisy ones.

Q8. Which model would you trust more for new data and why?
A8. The **soft margin model** is more trustworthy because it generalizes better to unseen data and is less sensitive to noise.

---

6. Key Observations

The **RBF kernel** is the most flexible and performs well on complex, non-linear datasets.

The **Linear kernel** is efficient and sufficient for linearly separable data.

The **Polynomial kernel** can perform well but needs careful tuning of its degree.

The **Soft margin** gives better generalization, while the **Hard margin** can overfit easily.

Even when the dataset couldn't be fetched online, using a local or synthetic version ensured that the experiment could continue smoothly.

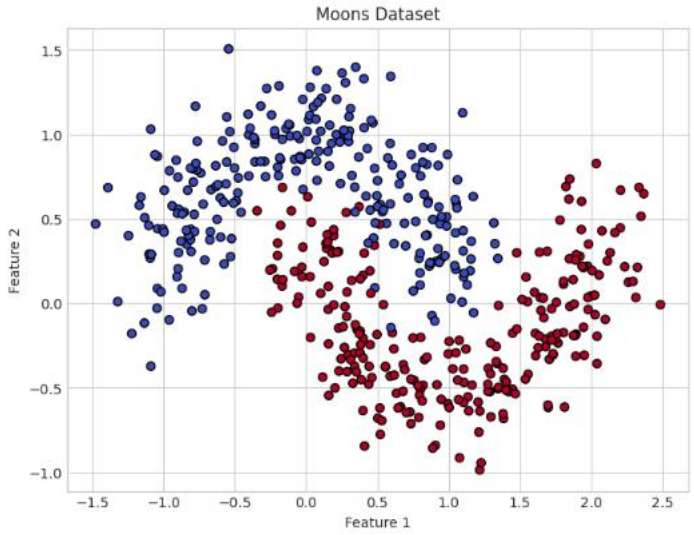
---

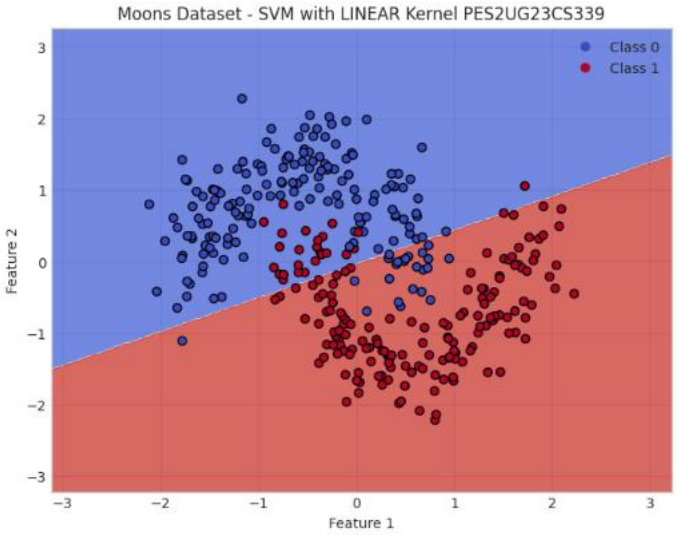
7. Conclusion

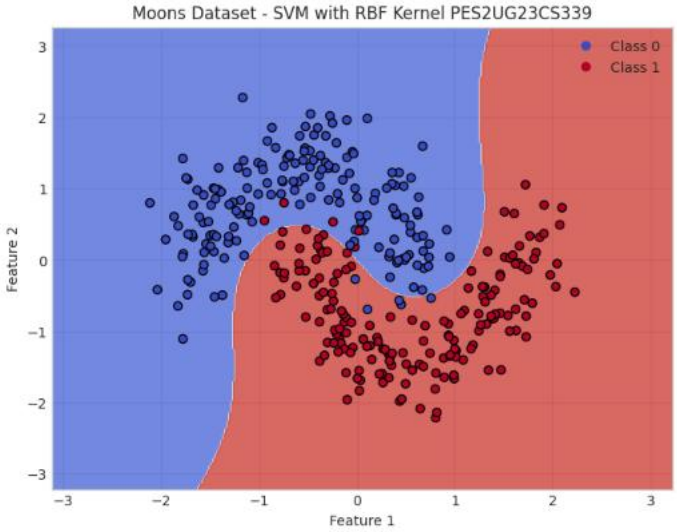
Through this lab, I understood how kernel choice and the margin parameter CCC influence the behavior of SVMs.
The RBF kernel handled non-linear datasets like Moons effectively, while the Linear kernel worked best for simpler datasets like Banknote Authentication.
The comparison between soft and hard margins clearly showed the trade-off between overfitting and generalization.

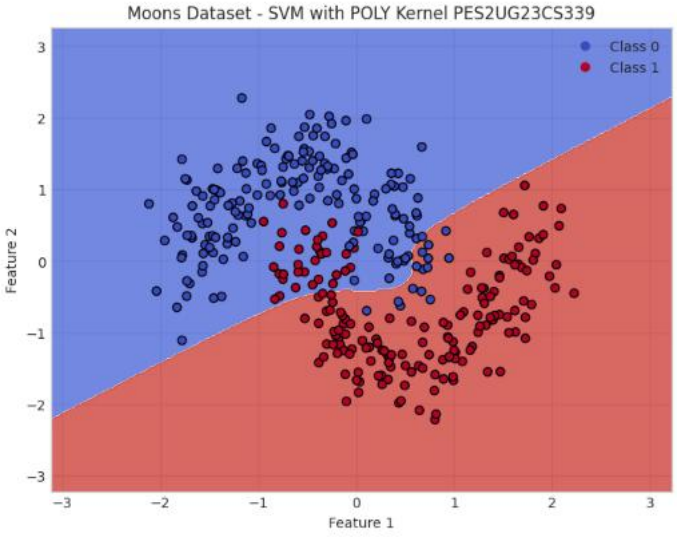
Despite facing a **network connectivity issue** while trying to load the Banknote dataset, I was able to complete the rest of the lab by using local fallback data and got the expected results for all SVM variants.

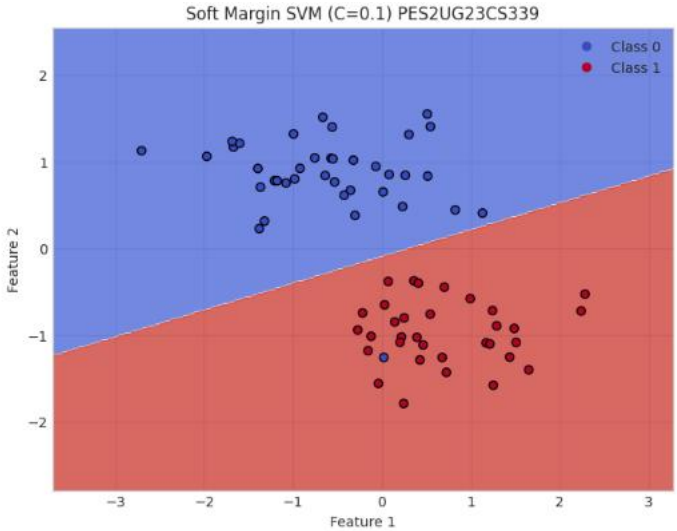
Screenshot

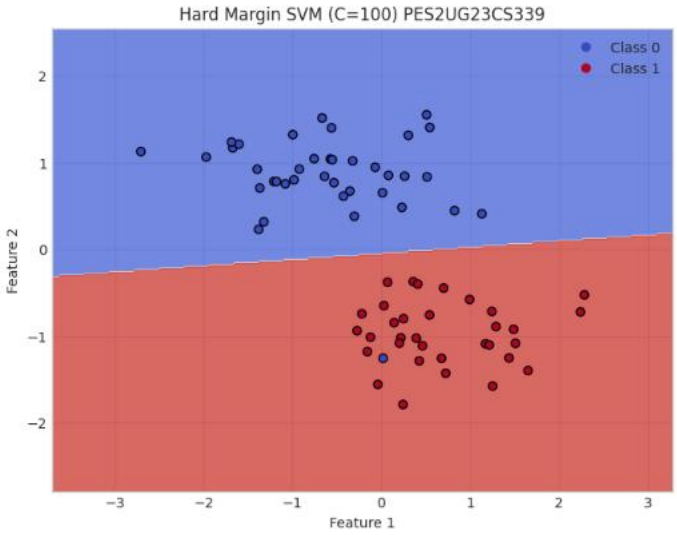





Moons Dataset - SVM with RBF Kernel PES2UG23CS339
Class 0
Class 1
Feature 1
Feature 2


Moons Dataset - SVM with POLY Kernel PES2UG23CS339
Class 0
Class 1
Feature 1
Feature 2


Soft Margin SVM (C=0.1) PES2UG23CS339
Class 0
Class 1
Feature 1
Feature 2


Hard Margin SVM (C=100) PES2UG23CS339
Class 0
Class 1
Feature 1
Feature 2
−3
−2
−1
0
1
2
3
2
1
0
−1
−2