A Report on

Support Vector Machines (SVMs) and Kernel Methods for Classification

Student Name: Sahith Reddy Pundru

Student ID: 24086678

Course: Machine Learning and Neural Networks

1. Introduction

Support Vector Machines (SVMs) are fundamental supervised learning algorithms widely employed for classification and regression tasks. Their main principle is to find a **hyperplane** that separates classes of data with the largest possible margin — the distance between the hyperplane and the closest data points from each class. These critical points, called **support vectors**, determine the orientation and position of the decision boundary.

Intuitive Analogy

Consider a tabletop scattered with apples and oranges. If the fruits form two groups that can be separated by a straight line, the task is simple. However, if the fruits form complex shapes, such as crescents or concentric circles, a linear boundary fails. Kernel functions address this problem by implicitly projecting data into **higher-dimensional feature spaces**, enabling linear separation in that transformed space. Kernels are therefore essential for capturing non-linear relationships in datasets.

2. Understanding Kernel Functions

Kernel functions map the input data into a higher-dimensional space without explicitly performing the transformation, a technique known as the **kernel trick**. By doing so, SVMs can efficiently handle non-linear patterns while keeping computation manageable.

Common Kernel Types

Kernel	Description	Use Case
Linear	Produces a straight-line decision boundary in the original feature space	Data is linearly separable
Polynomial	Generates curved boundaries; the degree parameter controls flexibility	Moderately non-linear datasets
RBF (Gaussian)	Maps data into an infinite-dimensional space to capture highly complex patterns	Highly non-linear datasets

Each kernel presents trade-offs between flexibility and the risk of overfitting. Proper selection depends on the structure of the data and the problem at hand.

3. Datasets for Demonstration

To illustrate how kernel choice influences SVM performance, three synthetic datasets were used:

Dataset	Structure	Complexity
Moons	Two crescent-shaped clusters	Non-linear
Circles	Concentric circular clusters	Highly non-linear
Linear	Two well-separated clusters	Linear

Dataset Details

- Each dataset consists of **300 samples**.
- The data was split so that **70% was used for training** the models, while the remaining **30% served for evaluation**.
- Noise was introduced to the Moons and Circles datasets to simulate real-world variability.
- Each dataset is **two-dimensional**, making it ideal for visualising SVM decision boundaries.

These datasets highlight the strengths and limitations of different kernels.

4. Methodology

A structured approach is adopted to analyze SVM behavior across different kernels.

4.1 Model Setup

1. **Linear SVM**
2. **Polynomial SVM** (degree = 3)
3. **RBF SVM** (gamma = 'auto')

Each model used identical training and test splits to ensure a fair comparison, while other hyperparameters were left at default values unless specified.

4.2 Evaluation Process

The models were trained on the majority of the dataset, reserving a portion for testing. **Accuracy** was calculated as the primary performance metric:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

Decision boundaries were visualized using meshgrids and contour plots, overlaying training and test points to evaluate classification effectiveness.

4.3 Visualisation

Decision boundaries were displayed using **color-coded regions** for predicted classes:

- **Training points:** circles with black edges
- **Test points:** squares with black edges
- **Background shading:** predicted class regions

Color palettes were chosen to ensure accessibility, including color-blind-friendly contrasts.

5. Results

5.1 Accuracy Summary

Dataset	Linear	Polynomial	RBF
Moons	0.90	0.83	0.92
Circles	0.42	0.42	0.92
Linear	0.96	0.90	0.97

5.2 Interpretation

Moons Dataset:

- Curved, non-linear structure.
- **RBF kernel** performs best due to its ability to model complex boundaries.
- Polynomial kernel is moderately effective, but slightly limited in flexibility.
- Linear kernel is less effective at capturing the crescent pattern.

Circles Dataset:

- Strongly non-linear and concentric.
- Linear and Polynomial kernels fail to separate classes effectively.
- **RBF kernel** correctly captures the circular decision boundary.

Linear Dataset:

- Linearly separable.
- Linear kernel achieves near-perfect accuracy.
- RBF kernel offers minimal improvement, which is not necessary but does not harm performance.

6. Figures (Descriptive)

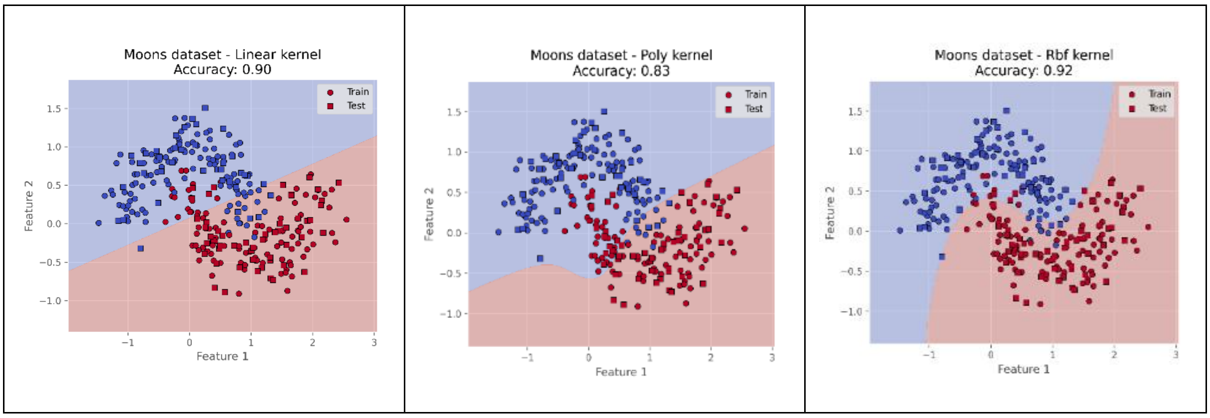

Figure 1 — Moons Dataset:

Decision boundaries for Linear, Polynomial, and RBF kernels. Demonstrates RBF's flexibility in handling crescent shapes.

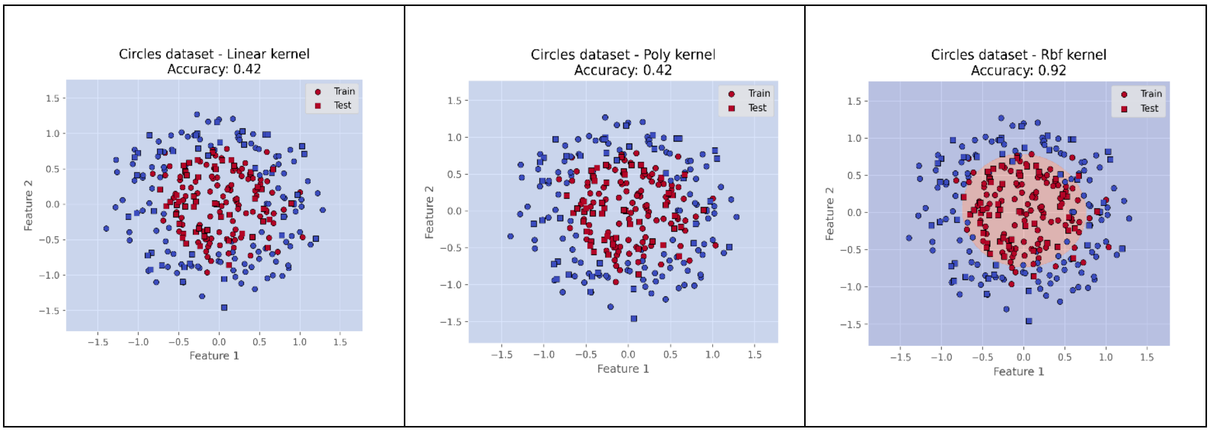

Figure 2 — Circles Dataset:

Linear and Polynomial kernels fail, whereas RBF performs optimally. Highlights the importance of choosing the right kernel.

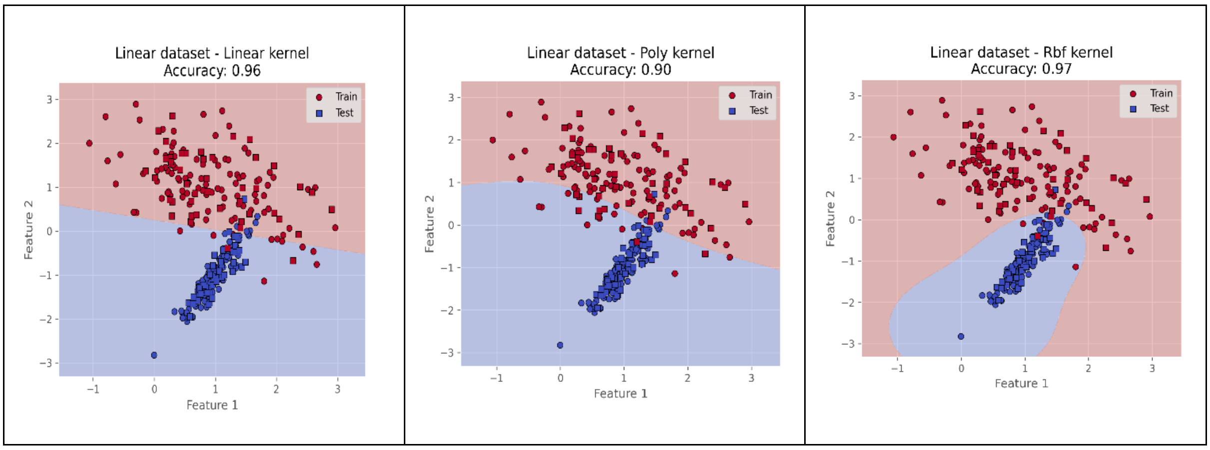

Figure 3 — Linear Dataset:

Linear kernel suffices; RBF slightly overfits but maintains high accuracy. Illustrates how simple data benefits from straightforward kernels.

7. Discussion

7.1 Key Insights

- Kernel selection strongly affects SVM performance.
- Linear kernels are effective for simple, linearly separable data.
- Polynomial kernels provide moderate flexibility but require careful tuning of the degree parameter.
- RBF kernels generalize well to complex non-linear patterns but require careful tuning of gamma.

7.2 Hyperparameter Considerations

- **C (regularization parameter):** balances margin maximization and misclassification tolerance.
- **Gamma (for RBF):** controls the influence of each training point; high gamma can lead to overfitting.
- **Polynomial degree:** higher values increase flexibility but may overfit noisy data.

7.3 Real-World Applications

- Medical diagnosis (e.g., tumor classification)
- Fraud detection in banking
- Image recognition and computer vision
- Text categorization
- Bioinformatics (e.g., gene expression classification)

7.4 Educational Value

Visualizing SVM decision boundaries aids in understanding:

- How support vectors define the hyperplane
- Effects of different kernels on non-linear separation
- Trade-offs between flexibility and overfitting

8. Conclusion

This tutorial explained:

- How SVMs operate using margin maximization
- The role of kernel methods in capturing non-linear patterns
- Evaluating and contrasting the Linear, Polynomial, and RBF kernel performances
- Importance of visualisation for interpreting model behavior

In conclusion, SVMs are highly effective for classification tasks. Choosing the appropriate kernel and tuning hyperparameters are crucial for achieving robust and interpretable models suitable for diverse applications.

References

1. Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 20, 273–297.
2. Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
3. Pedregosa, F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
4. Brownlee, J. (2020). A Gentle Introduction to Support Vector Machines in Python. Machine Learning Mastery.
5. Raschka, S. (2018). Python Machine Learning. Packt Publishing.
6. Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press.