How Kernel Choice Affects SVM Decision Boundaries

Name : Likith Rathipinni **Student id** : 24005943

Github link : https://github.com/likith-18-ai/svm-kernel-tutorial-report

Introduction

The Support Vector Machines (SVMs) are an essential approach to modern machine learning because they provide an excellent performance in terms of classification with a solid mathematical foundation. Although the concept of the canonical linear SVM is very simple, this approach has intrinsic limitations with regard to expressiveness. This limitation is alleviated by kernel functions which allow classification in higher-dimensional feature spaces without requiring explicit transformations. This tutorial discusses how the selection of the kernel affects the behavior of SVM using the visual examples and analysis. With the comparison of three main kernels, linear, polynomial and radial basis function (RBF), we question the strengths, restrictions, as well as the applicability to real-world data.

Objective of the report

One of the most crucial decisions that should be made during the training of an SVM model is the choice of the kernel. Kernels control decision boundary geometry and have a direct impact on performance, generalization, and interpretability. Most tutorials mention kernels in many cases but hardly any of them demonstrate their behavior in a manner that is visually understandable. The report supplements the conceptual knowledge, as it combines theory and graphical representations designed in Python.

Personal Insight

This tutorial significantly improved my knowledge of the role of the kernel functions in reference to the SVM behavior. In particular, testing the data on the Moons and Circles datasets served to elaborate on the differences between the linear, the polynomial and the RBF kernels in a simplified form. The fact that this method is practical in nature also contributed to making the concepts that previously seemed to be abstract understandable, especially the role of the parameters such as C and gamma.

Theoretical Background

1. Linear Kernel — The Baseline Case

The **linear kernel** computes a simple dot product between data points. It assumes that the classes can be separated by a straight line (or hyperplane in higher dimensions). When this assumption is true, the linear kernel provides:

- Excellent interpretability
- A stable model that does not overfit easily

But linear kernels fail when the data form nonlinear shapes such as spirals, moons, circles, or clusters.

2. Polynomial Kernel — Adding Controlled Nonlinearity

The **polynomial kernel** introduces interactions between features by raising the dot product to a specified degree. For example, a degree-3 polynomial kernel can model cubic curvature. This allows the SVM to represent:

- Curved boundaries

- Moderate nonlinear patterns

However, polynomial kernels can be sensitive to:

- High-degree polynomials (risk of overfitting)
- Poorly scaled features

They offer more flexibility than linear kernels but remain more interpretable than RBF kernels.

3. RBF (Gaussian) Kernel

The **Radial Basis Function (RBF)** kernel is the most widely used kernel in modern SVM applications because it can model extremely complex boundaries **(Burges, 1998)**. The RBF kernel measures similarity based on distance:

- Points close together in input space are considered similar
- Points far apart have little influence on each other

This leads to smooth, adaptive decision boundaries that wrap naturally around data clusters. While extremely powerful, RBF kernels reduce model interpretability , a known trade-off in machine learning.

4. Role of Hyperparameters: C and Gamma

Two hyperparameters fundamentally shape the behaviour of SVMs with nonlinear kernels:

C — Regularization Strength

The parameter **C** determines how strictly the SVM tries to avoid misclassifying points.

- **Low C → wider margin, softer boundary, better generalization**
- **High C → narrow margin, complex boundary, risk of overfitting**

C controls the balance between margin size and the classification accuracy.

Gamma — Influence of Individual Points

Gamma controls how much influence individual training samples have on the decision boundary.

- **Low Gamma → smooth, simple boundary**
- **High Gamma → tightly curved boundary that may fit noise**

Gamma is especially important for RBF kernels because it determines how rapidly similarity decays with distance.

Together, **C** and **gamma** define the shape and smoothness of the decision surface. Improper tuning can lead to overfitting or underfitting, which is why hyperparameter search is essential.

Mathematical calculations

1.SVM Objective Function (Soft-Margin Primal Form)

This is the optimization problem solved internally by the SVM classifier:

$$\min_{w,b,\xi} \frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{n} \xi_i$$

subject to:

$$y_i(w^\top x_i + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

- C= regularization parameter
- ξ_i= slack variables allowing misclassified or margin-violating points

This is the theoretical basis for how SVM finds the best separating hyperplane.

2. Margin Formula (Linear SVM)

For a linear decision boundary $w^\top x + b = 0$:

$$\text{Margin} = \frac{1}{\| w \|}.$$

3. SVM Decision Function (General Kernel Form)

This is the function used to make predictions:

$$f(x) = \sum_{i=1}^{n} \alpha_i \, y_i K(x_i, x) + b.$$

Final prediction:

$$\hat{y} = \text{sign}(f(x)).$$

Here:

- α_i= dual coefficients
- $K(\cdot,\cdot)$= kernel function
- x_i= support vectors

4. Common kernels (formulas)

- **Linear kernel:**

$$K(x, x') = x^\top x'.$$

- **Polynomial kernel** (degree d, coef $r$and scale γ):

$$K(x, x') = (\gamma x^\top x' + r)^d.$$

- **RBF (Gaussian) kernel:**

$$K(x, x') = \exp\left(-\gamma \| x - x' \|^2\right),$$

where $\gamma > 0$controls the kernel width (larger γ→ more local influence).

5. Role of Regularization Parameter $\boldsymbol{C}$

The SVM optimization balances two objectives:

- Minimize $\| w \|^2$→ maximize margin
- Minimize total slack $\sum \xi_i$→ reduce misclassifications

$$\min \frac{1}{2} \| w \|^2 + C \sum_i \xi_i$$

Therefore

- Large C: penalize errors → complex boundary
- Small C: allow some errors → smoother boundary

Role of RBF Parameter $\boldsymbol{\gamma}$

$$K(x, x') = e^{-\gamma \| x - x' \|^2}$$

- High γ: decision boundary bends tightly around points (overfitting)
- Low γ: boundary becomes smoother (underfitting risk)

6. Performance metrics

- **Accuracy:** $\text{accuracy} = \frac{\text{correct predictions}}{\text{total}}$
- **Confusion matrix** :

$$\begin{bmatrix} \text{TP} & \text{FN} \\ \text{FP} & \text{TN} \end{bmatrix}$$

- **Precision:** $\text{precision} = \frac{TP}{TP+FP}$
- **Recall :** $\text{recall} = \frac{TP}{TP+FN}$
- **F1 score:** $F1 = 2 \cdot \frac{\text{precision}\cdot\text{recall}}{\text{precision}+\text{recall}}$

Kernel Behavior and Decision Boundaries

The linear kernel offers interpretability and computational simplicity but fails to model curved relationships, as evident when applied to the Moons and Circles datasets. When the underlying structure bends or forms concentric shapes, the linear model cannot adapt. The polynomial kernel provides additional flexibility by modelling interactions of increasing degree .Yet, polynomial kernels may become unstable or overly sensitive to noise if the degree is not chosen carefully. In contrast, the RBF kernel provides exceptional flexibility and typically yields the best results for datasets with complex patterns. Its adaptive nature allows it to bend decision boundaries smoothly around data clusters, capturing subtle structures missed by other kernels. Through visual analysis, these distinctions become clear and form the basis for principled kernel selection.

Experiment 1: Moons Dataset (Linear vs Polynomial vs RBF)

Moons dataset is a nonlinear synthetic dataset hence offering a perfect testbed when studying the effect of different kernel functions on the Support Vector Machine behavior.

• **Linear kernel**: This creates a hyperplane that splits the data into hard barriers making it severely underfit since it is unable to act like the rounded moon shapes.

• **Polynomial kernel (degree 3):** a- little curvature is captured, and it is a better choice than the linear kernel, but it is not able to cope with the complicated shape.

• **RBF kernel**: Accurately models the moons with a smooth and flexible boundary, and has the best performance. The experiment has shown that choice of kernel has a significant influence on the perception of nonlinear structures of the model.

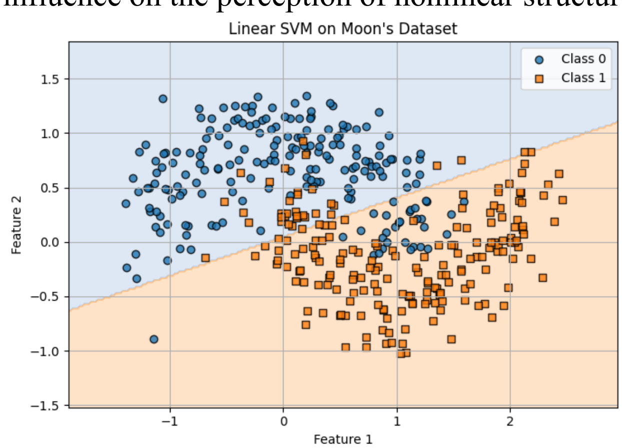


Figure 1 : The linear kernel produces a straight decision boundary, which fails to follow the curved structure of the moons, leading to misclassification.

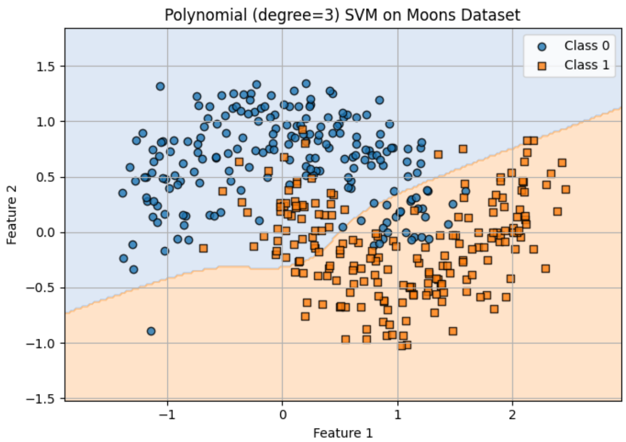


Figure 2 : The polynomial kernel introduces curvature and models the data better than the linear kernel, but still struggles with complex shapes.

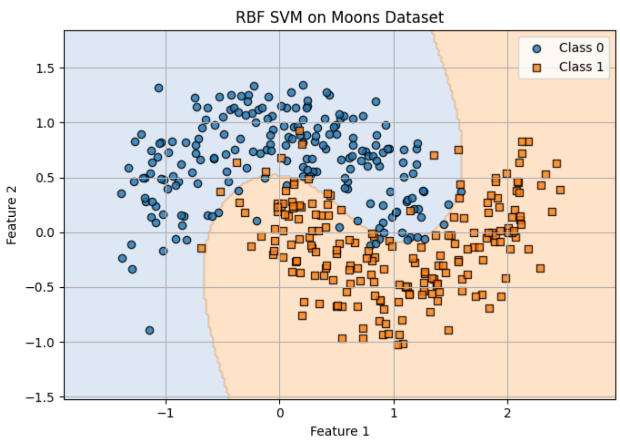


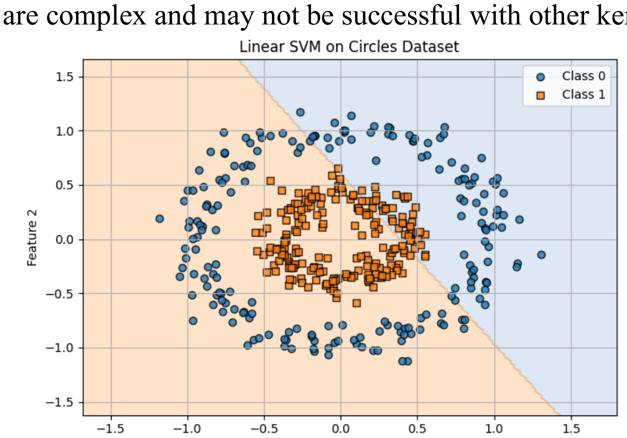
Figure 3 : The RBF kernel fits the moons effectively, capturing the nonlinear structure with a smooth and flexible decision boundary.

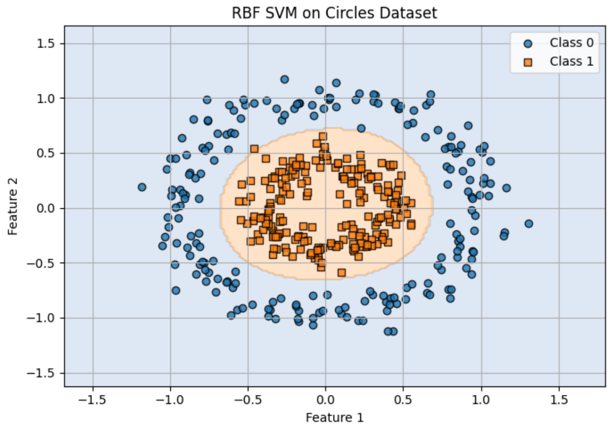
Experiment 2: Circles Dataset (Linear vs RBF)

The Circles data set is in the form of rings and is more difficult in simple decision boundaries.

• Linear kernel: Complete failure due to the inability to separate circular patterns by use of a straight line.

• RBF kernel: It has managed to capture the circular structure and is evidently superior to other kernels. This experiment underlines the fact that RBF kernel is very efficient in geometrical patterns that are complex and may not be successful with other kernels.



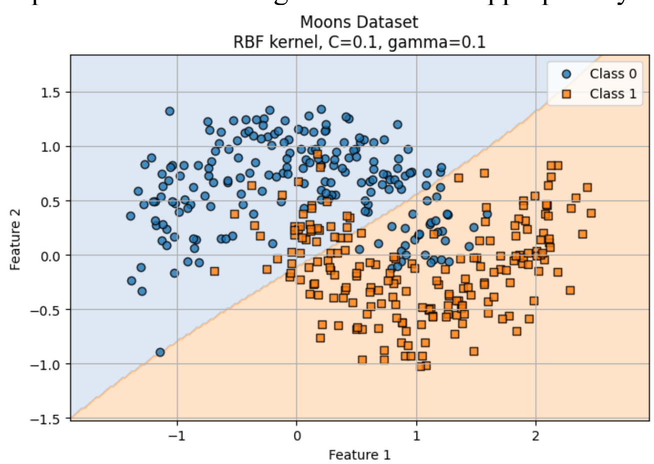


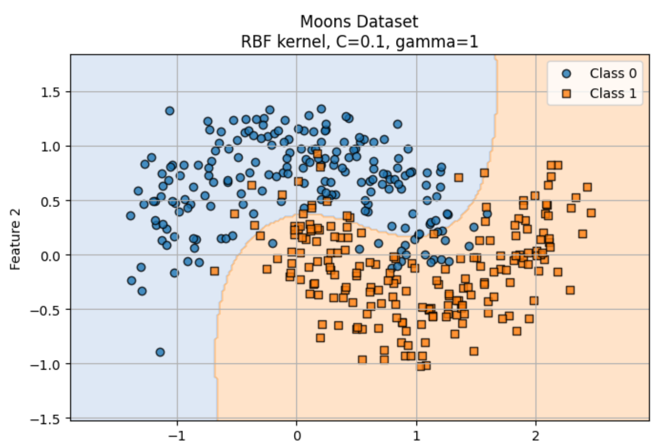
Experiment 3: Impact of Hyper parameters (C and Gamma on RBF)

Once the superiority of the RBF kernel has been established, this experiment assesses the behavior of the RBF kernel according to the hyperparameter settings.

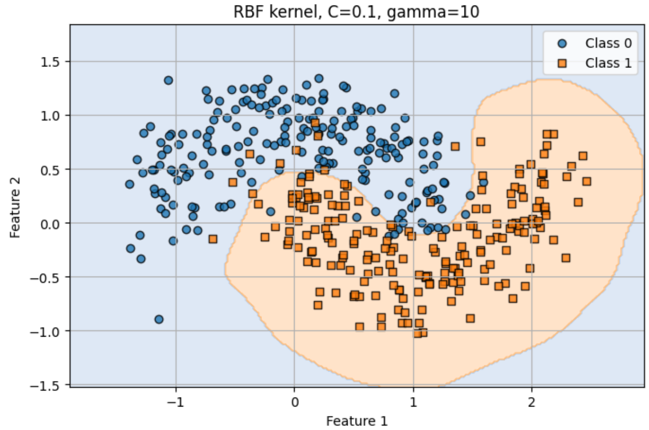
• **Gamma**: Small gamma values give smooth and simple boundaries that underfit and large gamma values give very tight boundaries that overfit.

• **C**: The low C values make the margins smoother, but can misclassify (underfit) the data; whereas the high C values make the model use the training data to fit the training data well, increasing the risk of overfitting. In this experiment, it has been demonstrated that balanced performance using a good kernel requires that the C and gamma be tuned appropriately.

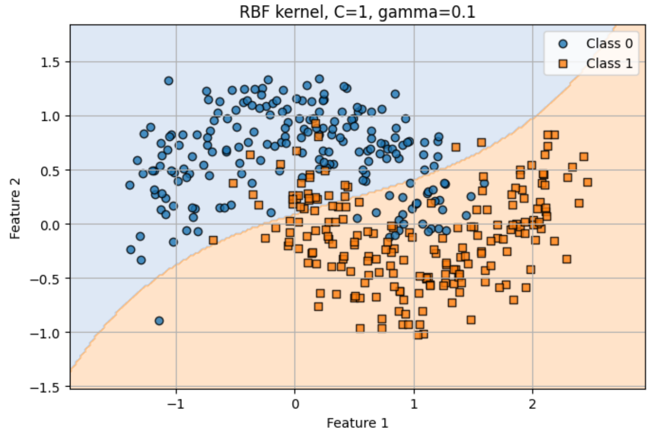




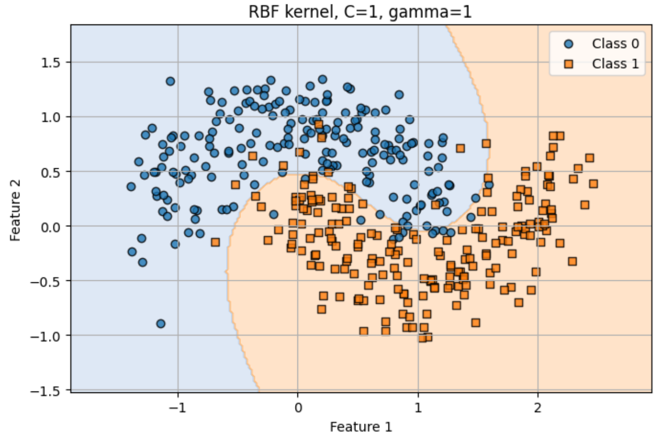
Moons Dataset
RBF kernel, C=0.1, gamma=10



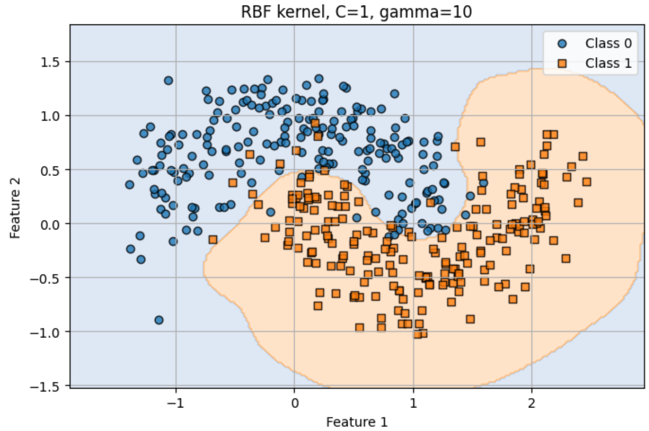
Moons Dataset
RBF kernel, C=1, gamma=0.1

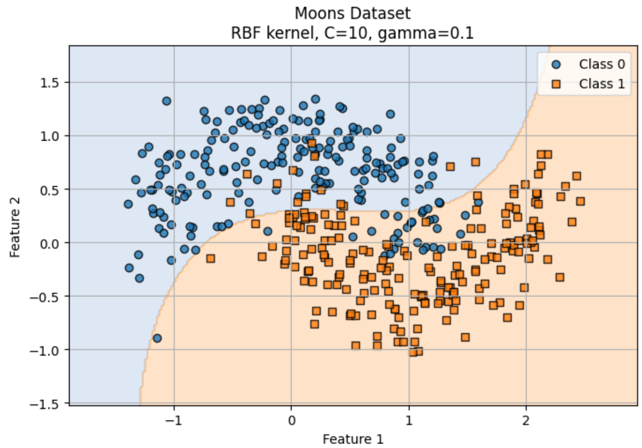


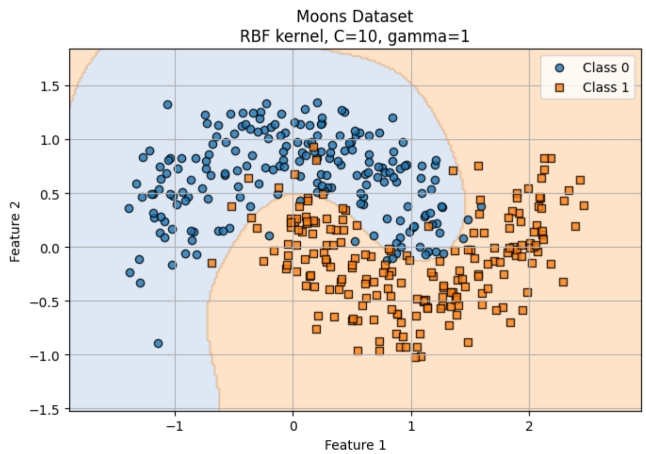
Moons Dataset
RBF kernel, C=1, gamma=1

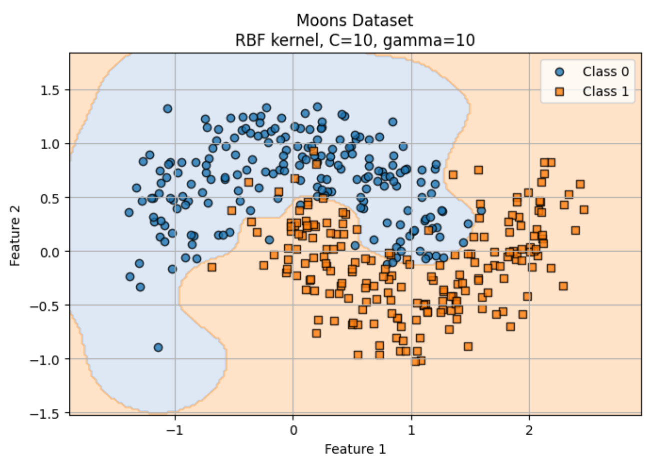


Moons Dataset
RBF kernel, C=1, gamma=10









Hyperparameter Effects

Two hyperparameters C and gamma present a strong dependence on the behavior of Support Vector Machine (SVM) models which use the polynomial or radial basis function (RBF) kernels. The regularization parameter C balances a trade off between the margin maximization and classification error minimization. The smaller the value of C, the smoother the decision boundary and the better the model generalization, but the higher the value of C, the more the predisposition of the model to overfitting. Reduced gamma results in more diffuse and smooth decision boundaries, increased gamma results in the model being responsive to single points, and generates complex decision boundaries that can be responding to noise, rather than real structure.

Practical Kernel Selection

It depends on the nature of data and the requirements of the problem being addressed that the kernel used. Linear kernel is best suited to high-dimensional, sparse data or in cases where interpretability is of the utmost importance. Polynomial kernels are useful when interactions among the features are based on preset degrees, or moderate non-linearity is anticipated. The RBF kernel is a strong default in the majority of applications due to its flexibility and capabilities with the heterogeneous data.

Ethical and Accessibility Considerations

Machine-learning systems are becoming embedded systems in decision-making systems that impact individuals and communities. More complicated models like the SVMs with RBF kernels can conceal the logic behind decision-making and therefore reduce transparency and accountability. Furthermore, overly flexible kernels can streamline the biases found in the training data, and so it is important to evaluate fairness. Accessibility is a very important issue in an academic presentation. Color-blind safe palettes, labelled axes and descriptive captions should be used in visualizations, and other markers and textual descriptions should be made available to meet the readership with different accessibility requirements.

Conclusion

Selection of Kernel is one of the pillars in the development of behavior of Support Vector Machines. The report has revealed how the linear, polynomial and RBF kernels create more and more flexible decision limits, both of which are appropriate in different contexts of data. Linear kernels are easy to understand and interpret, and polynomial kernels offer a tradeoff between predictive rigidity and elasticity, whereas RBF kernels offer the adaptive capability of an iron fist. The additional parameters of the model C and gamma can also be used to narrow behavior, and it is recommended to tune these parameters carefully in order to prevent overfitting. The full understanding of the working of the kernel will help practitioners create a better and more responsible SVM model in the real world. Combined with an interactive notebook depicting empirical illustrations, this report furnishes learners with conceptual and practical skills of SVM kernel selection.

References

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning*, 20*(3), 273–297.
→ Foundational paper introducing SVMs, maximum-margin classification, and kernel methods.
Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer.
→ Core textbook explaining SVM theory, decision boundaries, and kernel transformations.
Schölkopf, B., & Smola, A. J. (2002). Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*.* MIT Press.
→ Authoritative text on kernel functions, feature maps, and RBF/polynomial kernels.
Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2), 121–167.
→ Intuitive tutorial covering SVM geometry, kernels, and practical considerations.
Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830.
→ Reference for the library (scikit-learn) used to implement SVM models in your notebook.
scikit-learn Developers. (2024). Support Vector Machines — scikit-learn documentation.

https://scikit-learn.org/stable/modules/svm.html
→ Documentation source for SVC, kernel selection, hyperparameters C and gamma.