# An in-depth guide to support vector machines (SVM)

**Introduction to Support Vector Machines (SVM)**

One kind of supervised learning technique that is commonly used for classification and regression problems is Support Vector Machines (SVM). SVM stands out for its ability to handle datasets that are linearly and non-linearly separable using specialised functions called kernels. SVM's primary objective is to identify the best hyperplane that divides classes by the largest margin. This margin is computed as the distance between the hyperplane and the nearest data points, or support vectors ***(Cortes & Vapnik, 1995***).

**Why Use SVM?**

SVM is a popular choice for several reasons:

1. **Effective in High Dimensions:**  When there are a lot of features compared to samples, SVM works well. This makes it especially useful in text classification, image recognition, and bioinformatics tasks where data often has a high dimensionality ***(Schölkopf & Smola, 2002).***
2. **Versatile Kernels:** SVM can effectively use kernel functions to handle non-linearly separable data by converting it into a space with more dimensions. This adaptability allows it to solve a wide range of problems ***(Boser et al., 1992).***
3. **Robustness:**  In SVM, only a subset of training data known as support vectors affects the decision boundary. The model's resilience to noise and outliers is guaranteed by these support vectors ***(Cortes & Vapnik, 1995).***
4. **Clear Mathematical Foundation:** SVM is grounded in optimization theory, ensuring that the solution obtained is mathematically sound and globally optimal (at least for the linear case) ***(Vapnik, 1998)***.

**Applications of SVM:**

1. **Text Classification:** SVM is commonly used for spam email detection, sentiment analysis, and document categorization ***(Hastie et al., 2009)***.
2. **Image Classification:** In computer vision, SVM is employed for tasks like face detection and handwritten digit recognition ***(Schölkopf & Smola, 2002)***.
3. **Bioinformatics:** SVM helps classify proteins, predict diseases, and analyze gene expression data ***(Platt, 1999).***

**The Role of Kernels in SVM**

Kernels are the core of SVM's versatility. A kernel function transforms input data into a space with greater dimensions where each class can be efficiently split by a linear hyperplane ***(Schölkopf & Smola, 2002)***. The kernels allow SVM to tackle non-linear classification problems efficiently.

**Understanding Kernels**

In a higher-dimensional space, a kernel calculates the dot product between two points without figuring out the transformation directly. This procedure, referred to as the kernel technique, drastically lowers the processing cost ***(Boser et al., 1992).***

**Types of Kernels:**

1. **Linear Kernel:**
   * **Equation:** K(x,y)=xTy*K*(*x*,*y*)=*xTy*
   * Directly maps the data into the original feature space.
   * Fits data that is linearly separable.
   * Example: Classifying emails as spam or not based on word frequency ***(Hastie et al., 2009).***
2. **Polynomial Kernel:**
   * **Equation:** K(x,y)=(xTy+c)d*K*(*x*,*y*)=(*xTy*+*c*)*d*
   * Projects data into a polynomial feature space.
   * Controlled by the degree parameter (d*d*), which defines the polynomial's complexity.
   * Example: Classifying patterns with moderate non-linearity, such as distinguishing handwriting styles ***(Schölkopf & Smola, 2002).***
3. **Radial Basis Function (RBF) Kernel:**
   * **Equation:** K(x,y)=exp⁡(−γ∣∣x−y∣∣2)*K*(*x*,*y*)=exp(−*γ*∣∣*x*−*y*∣∣2)
   * Maps data into an infinite-dimensional space.
   * Controlled by the gamma parameter (γ*γ*), which determines the influence of individual data points.
   * Example: Detecting faces in images ***(Platt, 1999).***
4. **Sigmoid Kernel:**
   * **Equation:** K(x,y)=tanh⁡(αxTy+c)*K*(*x*,*y*)=tanh(*αxTy*+*c*)
   * Mimics the activation function in neural networks.
   * Less commonly used due to its tendency to behave like RBF under certain conditions ***(Hastie et al., 2009)***.

**Dataset Overview**

The dataset used in this tutorial is the Iris dataset, a benchmark dataset for classification tasks ***(UCI Machine Learning Repository, n.d.).*** It is simple yet versatile, making it ideal for learning machine learning techniques like SVM.

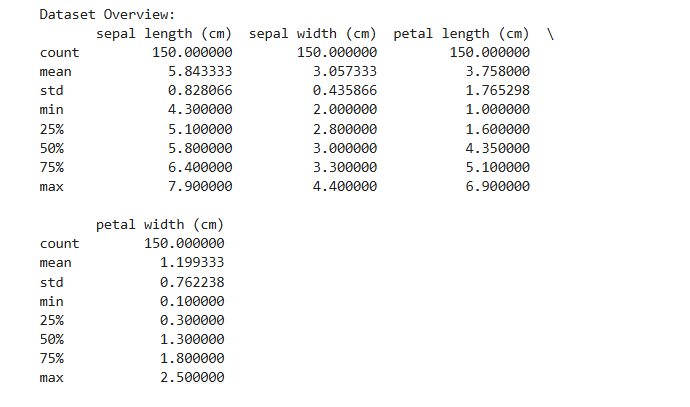
**Dataset Characteristics:**

1. **Number of Samples:** 150
2. **Classifications:** Three species of Iris flowers:
   * Setosa
   * Versicolor
   * Virginica
3. **Characteristics:**
   * Sepal length
   * Sepal width
   * Petal length
   * Petal width
4. **Objective Variable:** Species of the Iris flower.

**Objective:**

Task is to classify the Iris samples into one of the three species based on the given features.

****

****

**Data Preprocessing**

An integral part of any machine learning pipeline is data preprocessing. It is particularly crucial for SVM since the method is sensitive to the size of the input characteristics.

**Why is Preprocessing Necessary?**

1. **Feature Scaling:** Without scaling, features with large ranges can dominate the optimization process, skewing the results. Standardization ensures all features contribute equally to the model ***(Hastie et al., 2009).***
2. **Data Splitting:**  To evaluate the model's performance, the dataset is divided into training (70%) and testing (30%) subsets ***(UCI Machine Learning Repository, n.d.).***

**Exploring Different Kernels**

**Objective:**

Compare the performance of SVM models using Linear, Polynomial, and RBF kernels.

**Hyperparameter Tuning:**

1. **C (Regularization Parameter):**
   * Strikes a balance between minimising classification error and optimising margin.
   * A high C*C* value results in a complex decision boundary, potentially overfitting.
   * A low C*C* value simplifies the decision boundary, possibly underfitting ***(Platt, 1999).***
2. **Degree:**
   * Only works with polynomial kernels.
   * Determines the complexity of the polynomial function.
3. **Gamma:**
   * Applicable for RBF and Polynomial Kernels.
   * A high γ*γ* value focuses on close neighbors, leading to overfitting.
   * A low γ*γ* value captures global patterns, resulting in underfitting ***(Hastie et al., 2009).***

**A screenshot of a computer program

Description automatically generated**

**Performance Visualization**

Confusion matrices reveal information about the model's performance. They display the quantity of misclassifications (off the diagonal) and accurate predictions (on the diagonal).

**A computer screen shot of a computer code

Description automatically generated**

**A diagram of a graph

Description automatically generated with medium confidence**

**A graph with blue squares and white text

Description automatically generated**

**A blue squares with white text

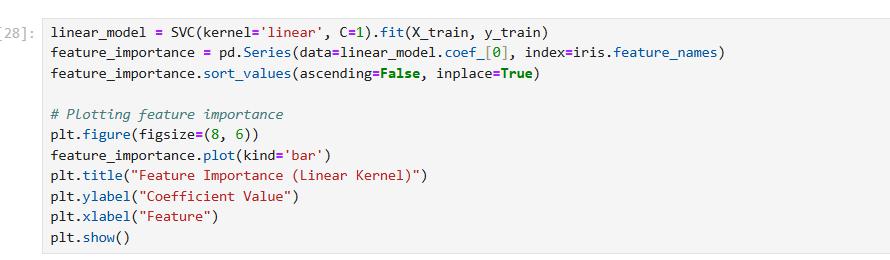
Description automatically generated**

**Feature Importance for Linear Kernel**

For linear SVM, feature importance can be determined using the coefficients of the hyperplane.

**Insights:**

1. Features with higher coefficients contribute more to the decision boundary.
2. For the Iris dataset, **petal length** and **petal width** are more influential than sepal features.

****

**A graph with blue rectangles

Description automatically generated**

**Support Vectors and Their Visualization**

**Support Vectors: What Are They?**

Support vectors are the data points that are nearest to the decision border. They establish the hyperplane's position and orientation.

**Visualization:**

1. **Dimensionality Reduction:** PCA reduces the dataset to two dimensions for visualization.
2. **Scatter Plot:** Focusses attention to support vectors and illustrates how they affect the decision boundary.

****

**A diagram of different colored circles

Description automatically generated**

**Sensitivity Analysis for Hyperparameters**

**Objective:**

To study how C*C* and γ*γ* influence the performance of the RBF kernel.

**Methodology:**

1. Evaluate the model for various combinations of C*C* and γ*γ*.
2. Use a heatmap to visualize the accuracy.

**Insights:**

1. **High C*C*:**
   * Overfits the data, leading to a complex decision boundary.
2. **Low C*C*:**
   * Simplifies the decision boundary but risks underfitting.
3. **High γ*γ*:**
   * Focuses on individual data points, potentially overfitting.
4. **Low γ*γ*:**
   * Captures broader patterns, resulting in smoother decision boundaries.

****

**A screenshot of a graph

Description automatically generated**

**Key Insights**

1. **Kernel Comparison:**
   * Linear kernels are efficient for linearly separable data.
   * Polynomial and RBF kernels are better suited for non-linear data.
   * RBF kernels often outperform others due to their flexibility.
2. **Feature Importance:**
   * Petal features play a dominant role in classifying Iris flowers.
3. **Hyperparameter Sensitivity:**
   * Proper tuning of C*C* and γ*γ* is essential for achieving optimal performance.
4. **Support Vectors:**
   * Only a subset of data influences the decision boundary, making SVM efficient.

**Conclusion**

By understanding core elements like kernels, support vectors, and hyperparameter tuning, SVM can be applied effectively to diverse datasets. This tutorial emphasizes the significance of preprocessing, kernel selection, and model evaluation, empowering users to harness SVM's potential ***(Cortes & Vapnik, 1995; Schölkopf & Smola, 2002).***

**References**

1. Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3), 273-297.
2. Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144-152.
3. Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Science & Business Media.
4. Schölkopf, B., & Smola, A. J. (2002). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press.
5. Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods - Support Vector Learning*, MIT Press.
6. UCI Machine Learning Repository. Iris Dataset. Retrieved from <https://archive.ics.uci.edu/ml/datasets/iris>.