Support Vector Machine: SVM

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both regression or classification challenges. However, it is mostly used in classification problems.

Support Vector Machine: SVM

1. Classification: In classification, SVM tries to find the hyperplane that best separates different classes in the feature space. This hyperplane is chosen in such a way that it maximizes the margin between the classes, which helps improve the generalization ability of the model.

2. Regression: SVM can also be used for regression tasks. In this case, it tries to find a hyperplane that best fits the data, while still maximizing the margin.

SVM: Key Concepts

1. Hyperplane: In a 2-dimensional space, a hyperplane is a line that separates the classes. In a higher-dimensional space, it's a subspace.

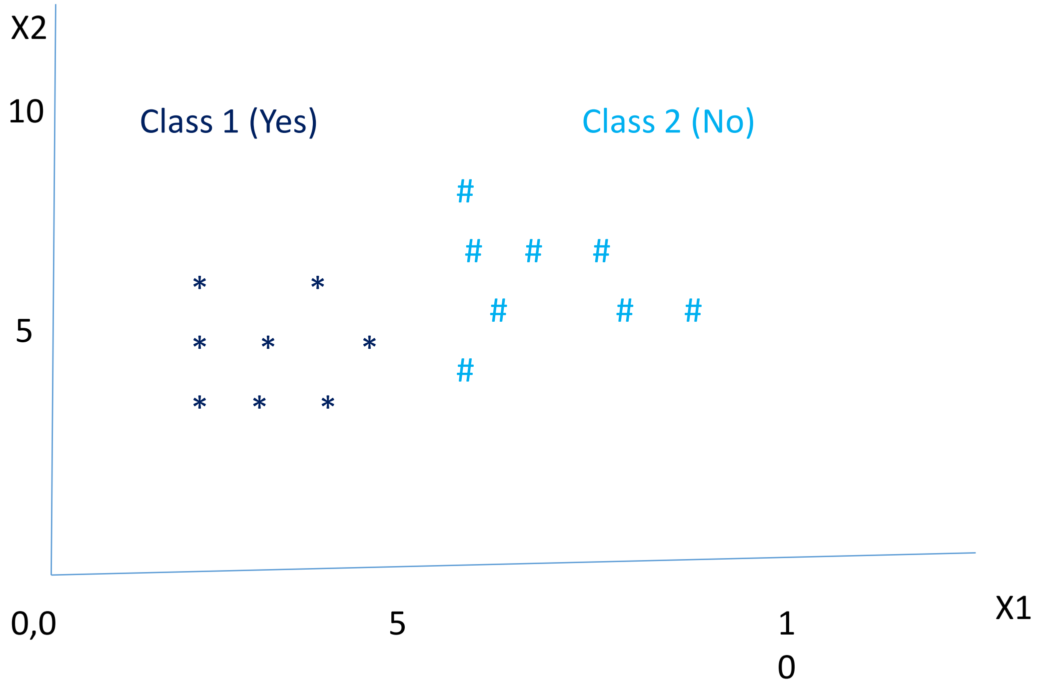
2. Support Vectors: These are the data points closest to the hyperplane and have a direct influence on the position and orientation of the hyperplane.

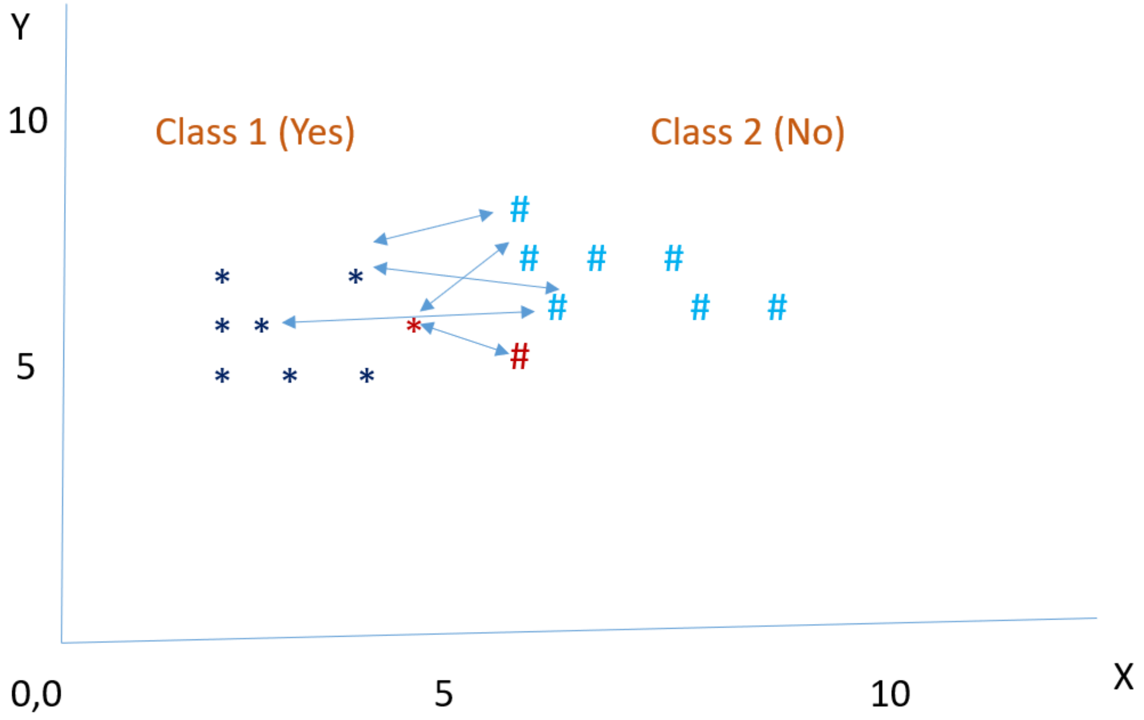
3. Kernel Trick: SVM can efficiently perform a non-linear classification/regression by mapping the original data into a higher-dimensional feature space. This is done using a kernel function, which computes the dot product in this higher-dimensional space. Common kernels include linear, polynomial, and radial basis functions (RBF).

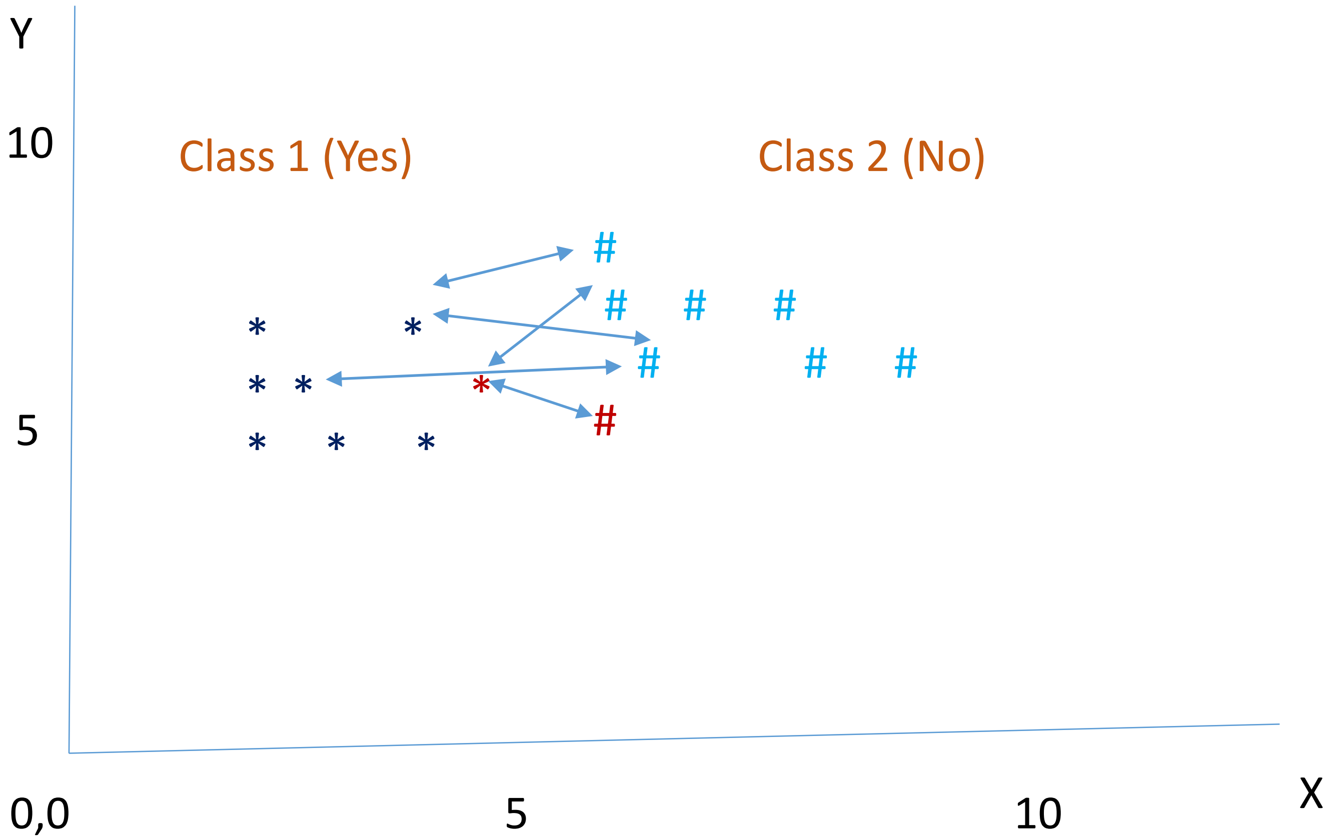
4. Regularization Parameter (C): It controls the trade-off between maximizing the margin and minimizing the classification error. A small C encourages a larger margin, while a larger C allows for fewer misclassifications.

If you're dealing with a dataset where you believe the classes are well-separated and there is not much noise, using a larger value of C might lead to better performance. On the other hand, if the dataset is noisy or there's overlap between classes, using a smaller C to encourage a larger margin might lead to better generalization.

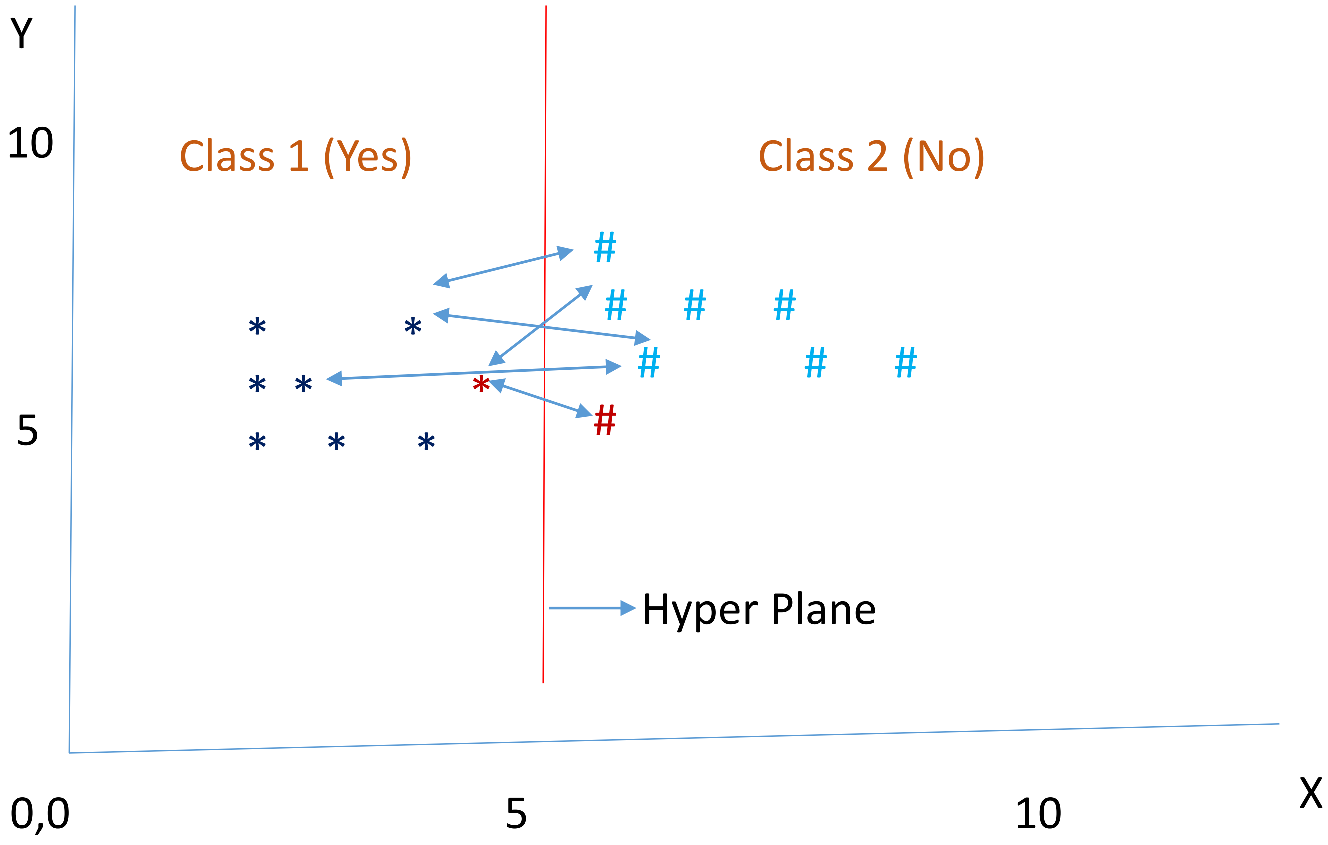
Let’s see…



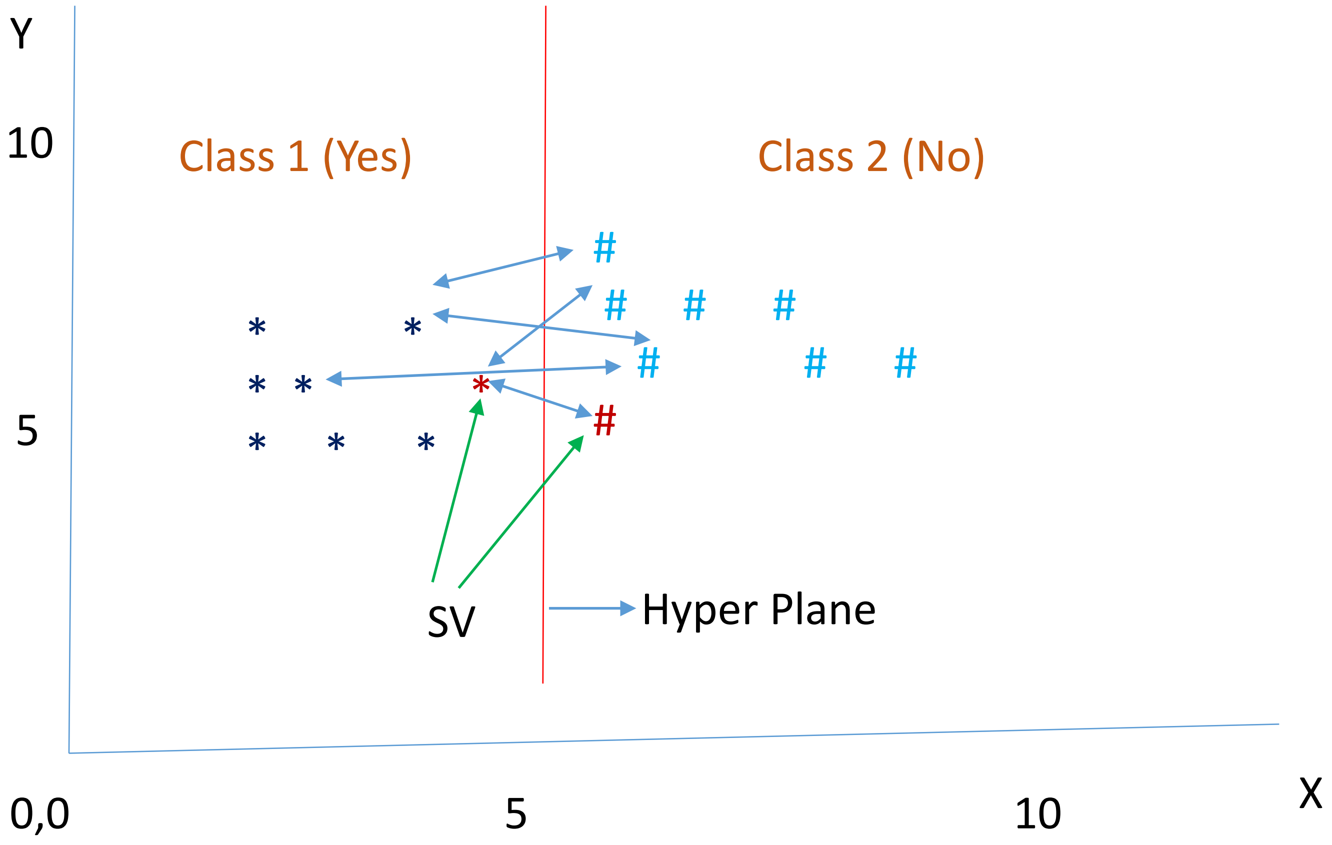

Y
10
5
Class 1 (Yes)
Class 2 (No)
0,0
5
10
X


Y
10
Class 1 (Yes)
Class 2 (No)
5
0,0
5
10
X

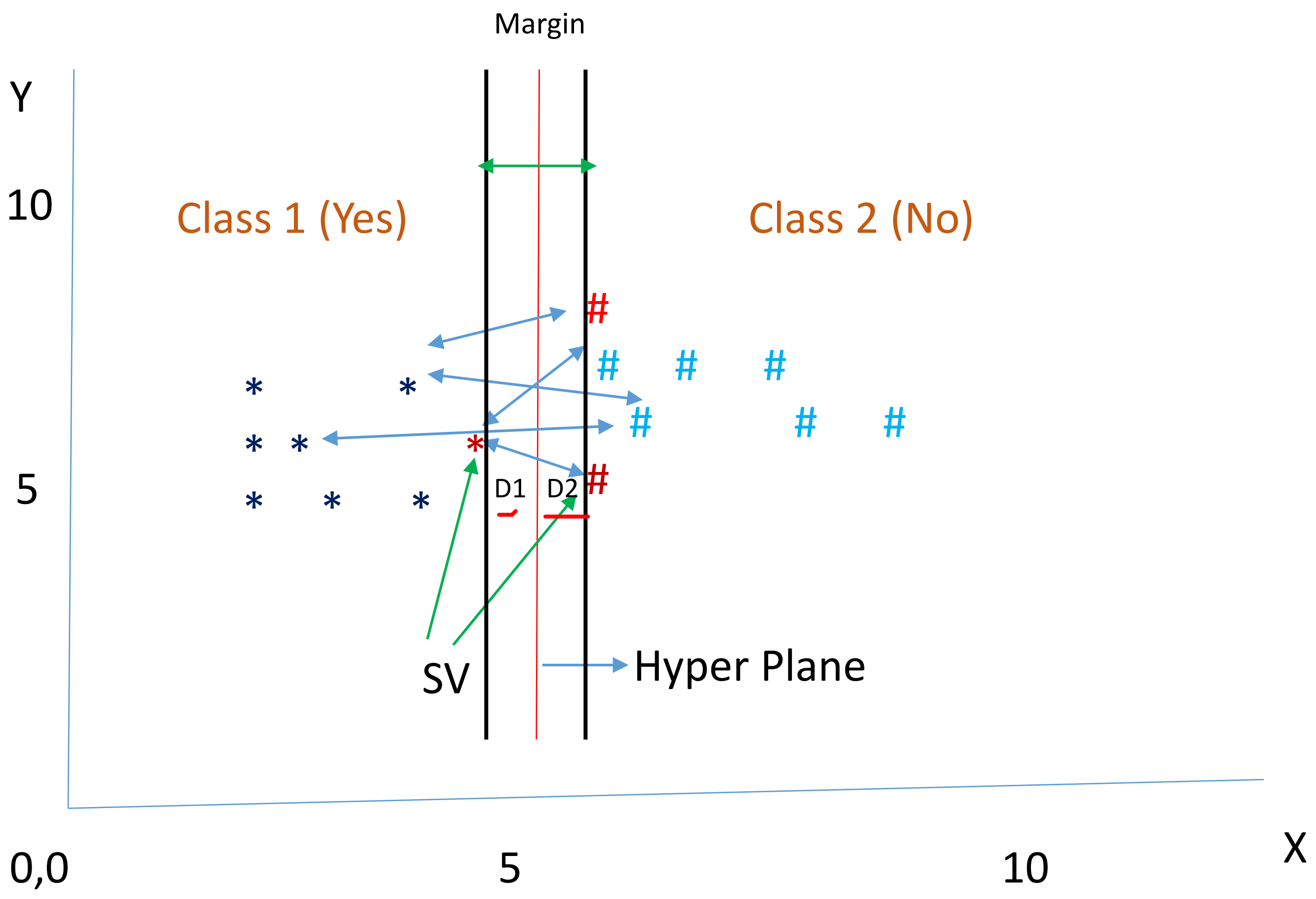
Support Vector Machine



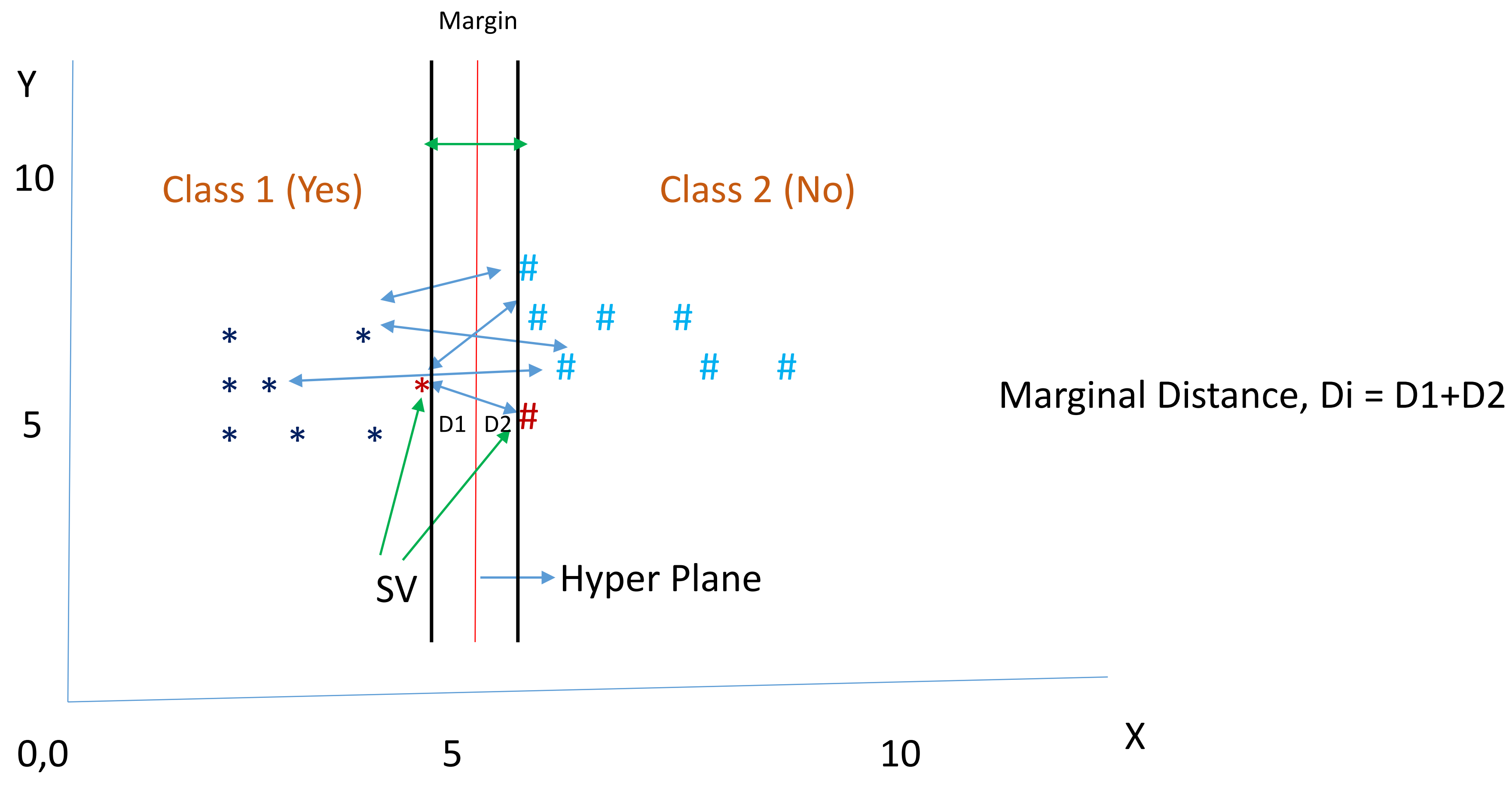
Support Vector Machine



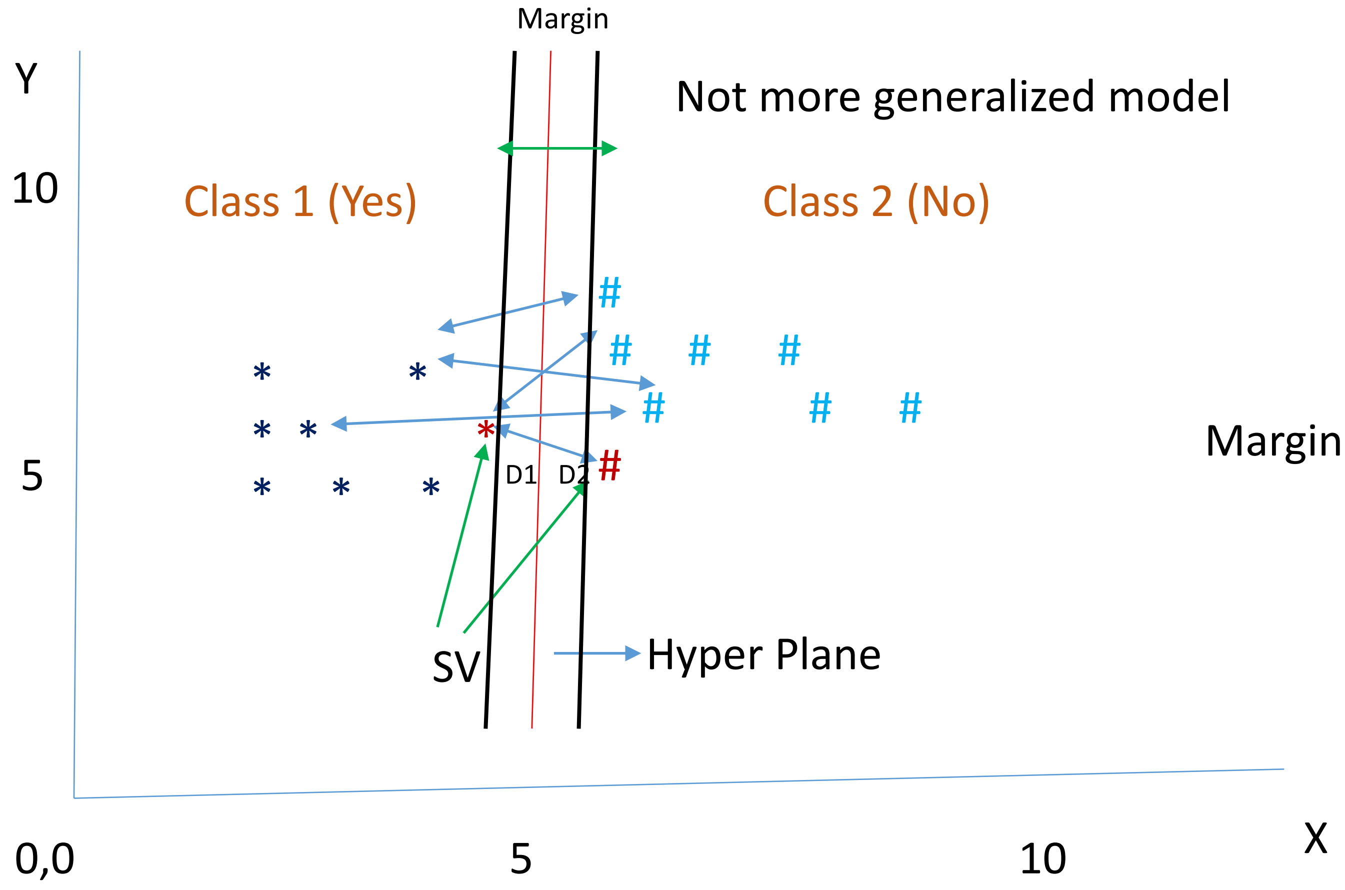
Support Vector Machine



Support Vector Machine



Support Vector Machine



Not more generalized model

Marginal Distance, Dj = D1+D2

Di >> Dj

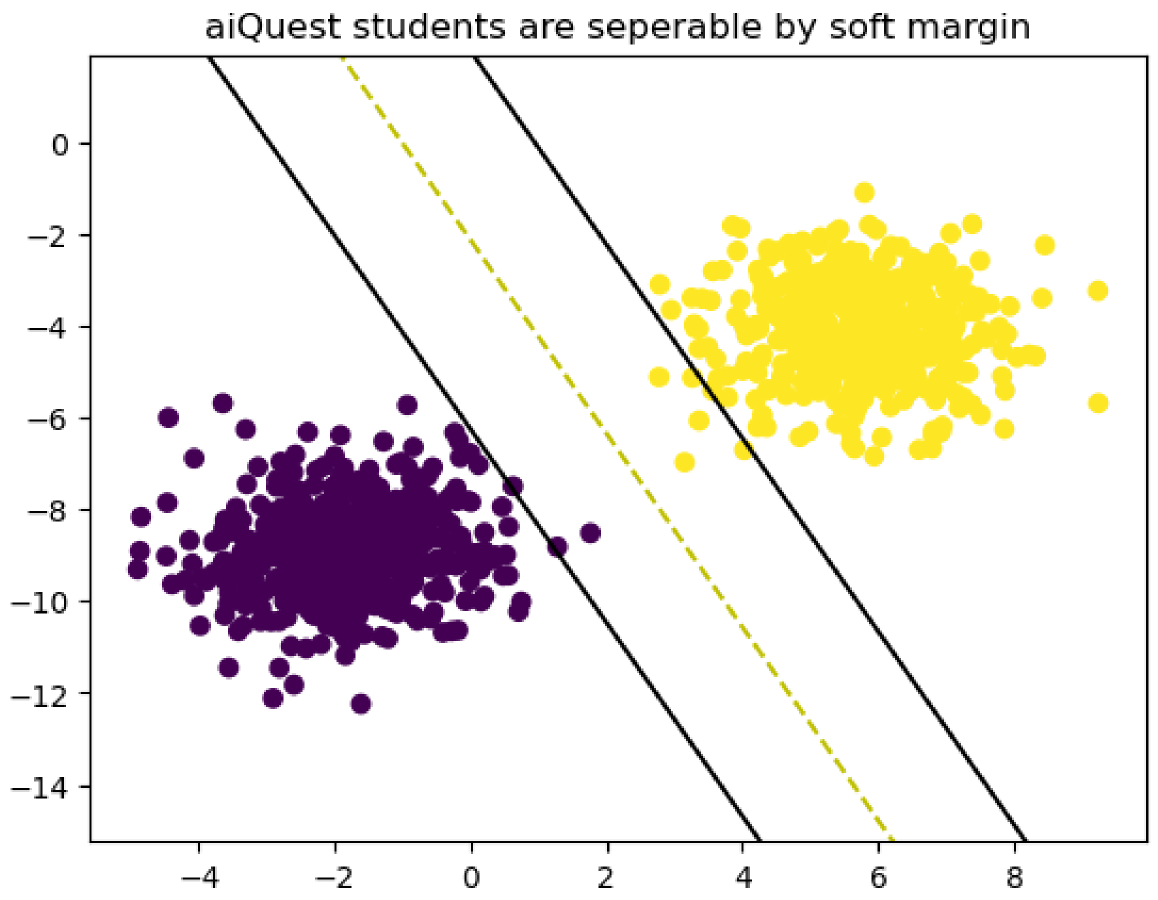
The Role of Margins in SVMs:

Sometimes, the data is linearly separable, but the margin is so small that the model becomes prone to overfitting or being too sensitive to outliers. Also, in this case, we can opt for a larger margin by using soft margin SVM in order to help the model generalize better.

Hard vs Soft Margin

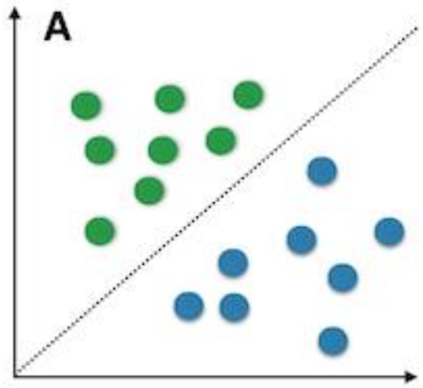
- When the data is linearly separable, and we don’t want to have any misclassifications, we use SVM with a hard margin.

- When a linear boundary is not feasible, or we want to allow some misclassifications in the hope of achieving better generality, we can opt for a soft margin for our classifier.

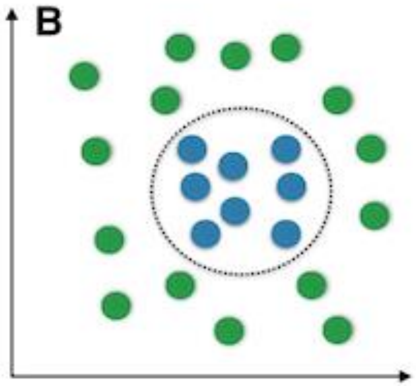


Support Vector Machine

Linearly Separable



Non-Linearly Separable

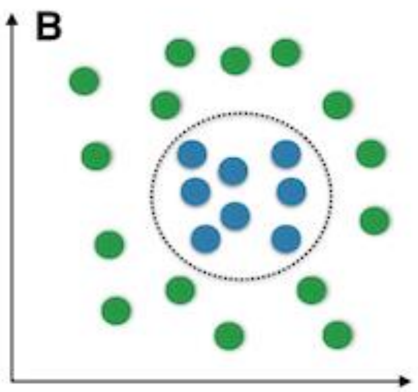


Support Vector Machine

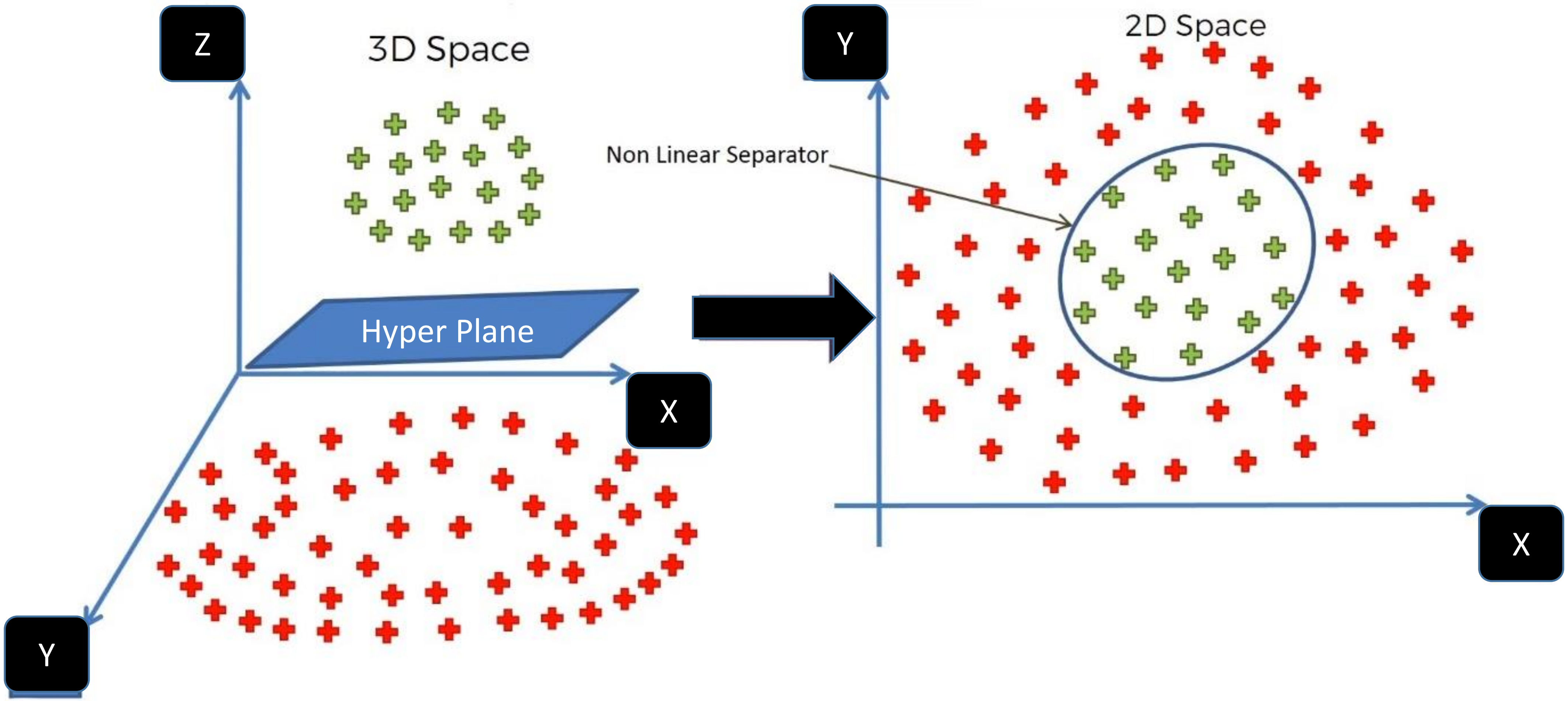
SVM Kernels Trick: Non-Linear SVM

Low Dimension to High Dimensions
2D to Higher Dimensions

Non-Linearly Separable



Support Vector Machine



Support Vector Machine

The most commonly used kernels in SVMs include:

1. Linear Kernel: $K(x, y) = x^T y$
2. Polynomial Kernel: $K(x, y) = (x^T y + c)^d$
3. Radial Basis Function (RBF) or Gaussian Kernel: $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$
4. Sigmoid Kernel: $K(x, y) = \tanh(\alpha x^T y + c)$

Support Vector Machine

The linear kernel is a type of kernel function used in Support Vector Machines (SVMs) and other machine learning algorithms. Unlike non-linear kernels, the linear kernel represents a linear transformation of the input features. The linear kernel is mathematically defined as:

$$K(x, y) = x^T y$$

Here:

- x and y are input feature vectors.
- $x^T y$ represents the dot product of the two vectors.

Support Vector Machine

When dealing with **non-linear data**, choosing an appropriate kernel is crucial for the success of Support Vector Machines (SVMs). The choice depends on the specific characteristics of your data, and different kernels may perform better in different scenarios. Here are some common non-linear kernels and considerations for their use:

1. **Radial Basis Function (RBF) or Gaussian Kernel:**
   - **Formula:** $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$
   - **Characteristics:** The RBF kernel is versatile and often a good choice for non-linear data. It introduces a notion of similarity based on the distance between data points in a high-dimensional space. The parameter σ controls the width of the Gaussian distribution, influencing the smoothness of the decision boundary.

Support Vector Machine

2. **Polynomial Kernel:**
   - **Formula:** $K(x, y) = (x^T y + c)^d$
   - **Characteristics:** The polynomial kernel allows SVMs to capture polynomial relationships between features. The degree (d) of the polynomial and a constant term (c) are hyperparameters that can be tuned. Higher degrees allow the model to capture more complex non-linearities, but they may also lead to overfitting.
3. **Sigmoid Kernel:**
   - **Formula:** $K(x, y) = \tanh(\alpha x^T y + c)$
   - **Characteristics:** The sigmoid kernel is another option for non-linear data. It can be seen as a simple model of a two-layer perceptron neural network. The hyperparameters α and c control the shape of the decision boundary.

Choose Best Kernel for SVM

Choosing the best kernel involves experimentation and, in many cases, cross-validation. Here are some considerations:

- **Data Exploration:** Understand the nature of your data and how different features interact. This understanding can guide the choice of a kernel that best captures the underlying relationships.
- **Model Complexity:** Consider the complexity of the decision boundary needed to model the data. RBF kernels are often a safe choice, but polynomial kernels with an appropriate degree may be effective as well.
- **Hyperparameter Tuning:** The performance of non-linear kernels depends on hyperparameters like σ, d, c, and α. Use techniques like grid search and cross-validation to find the optimal values for these parameters.