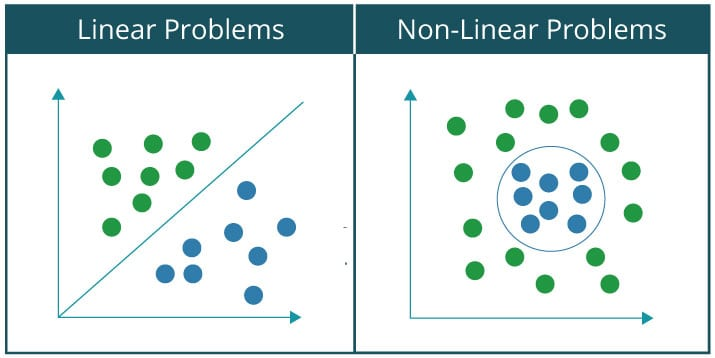
A Support Vector Machine (SVM) is a supervised machine learning algorithm used for classification and regression tasks. It's a powerful and versatile algorithm that is particularly well-suited for tasks with complex decision boundaries.

SVM finds hyperplane that best separates the data into different classes. The main goal is to maximize the margin, which is the distance between the hyperplane and the nearest data points from both classes

They can be broadly categorized into two main types: linear SVMs and nonlinear SVMs.



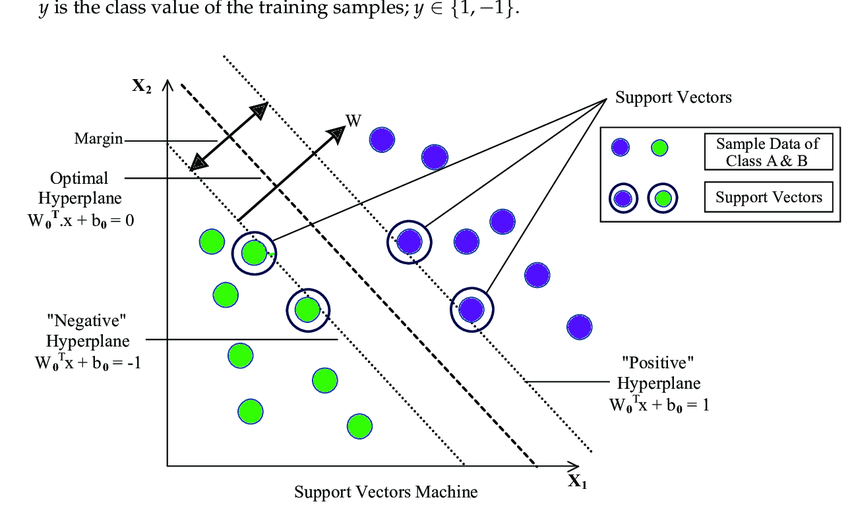
1. **Linear SVM**:

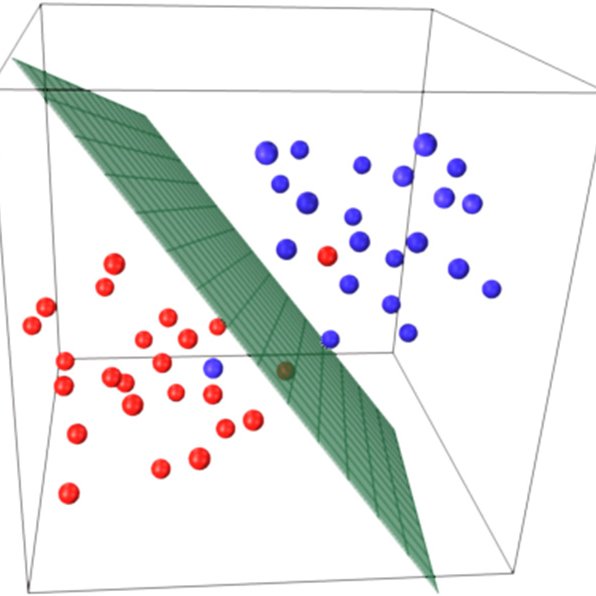
* Linear SVM is used when the data is linearly separable, which means you can draw a straight line (in 2D), a plane (in 3D), or a hyperplane (in higher dimensions) to separate the data into distinct classes.
* The goal of linear SVM is to find the hyperplane that maximizes the margin between the two classes. This hyperplane is chosen in such a way that it is equidistant from the nearest data points of both classes.
* Linear SVMs are computationally efficient and work well when the data is approximately linearly separable. They are widely used for tasks like text classification and simple image classification.

1. **Nonlinear SVM**:

* Nonlinear SVM is used when the data is not linearly separable, meaning you cannot separate the classes using a straight line or hyperplane in the input feature space.
* To handle nonlinear data, nonlinear SVMs use a technique called the kernel trick. The kernel trick allows SVMs to map the input data into a higher-dimensional space where it becomes linearly separable. Common kernel functions include the polynomial kernel, radial basis function (RBF) kernel, and sigmoid kernel.
* By transforming the data into a higher-dimensional space and finding a hyperplane there, nonlinear SVMs can capture complex decision boundaries in the original feature space.
* Nonlinear SVMs are more flexible than linear SVMs and can handle a wider range of classification problems.







Main concepts need to remember:

* Support vectors
* Hyperplane
* Margin
* Kernel trick
* C parameter
* Soft margin & Hard margin
* Multiclass classification

**Support Vectors:** In SVM, data points are plotted in a multidimensional space, and a hyperplane is chosen to separate the data into different classes. Support vectors are the data points closest to the hyperplane and are crucial for defining the decision boundary.

The hyperplane in an SVM is determined by the support vectors. More specifically, the weight vector (w) and the bias term (b) are computed based on these support vectors. Other data points, which are not support vectors, do not affect the hyperplane's position.

Points that are inside the margin but still correctly classified are called "slack" support vectors.

**Hyperplane:** In a binary classification problem (two classes), the hyperplane is the decision boundary that maximizes the margin between the two classes. SVM aims to find the hyperplane that best separates the data.

w⋅x+b=0

w is the weight vector (or normal vector) perpendicular to the hyperplane. It determines the orientation of the hyperplane.

b is the bias term (also called the offset or intercept), which shifts the hyperplane away from the origin along the direction of the weight vector.

x is the feature vector of a data point.

**Margin:** The margin is the distance between the hyperplane and the nearest data point (support vector) from each class. SVM tries to maximize this margin, which leads to better generalization.

the margin is inversely proportional to the magnitude (Euclidean norm) of the weight vector (∥w∥) of the hyperplane. Specifically, the margin (M) can be expressed as:

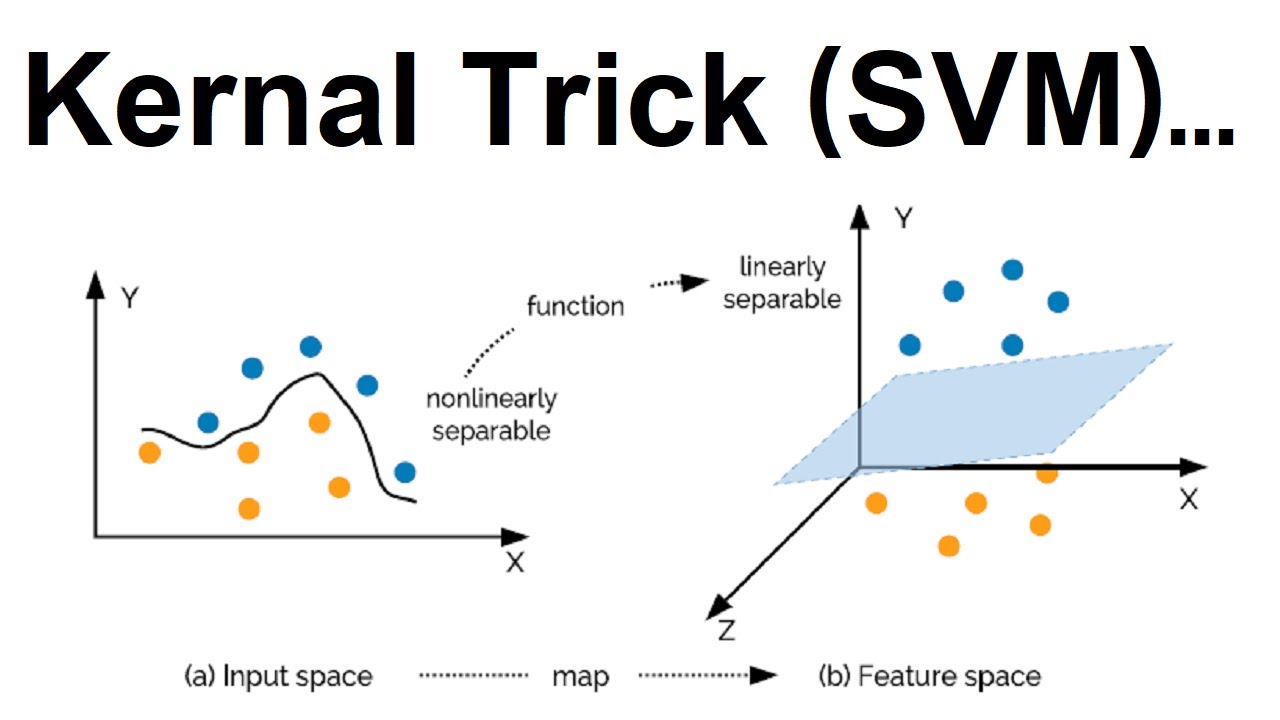
M = 2/∥w∥

Here, the factor of 2 is included for convenience.

w is the weight vector (normal vector) perpendicular to the hyperplane.

∥w∥ is the magnitude (length) of the weight vector.

**Kernel Trick:** In cases where a linear hyperplane cannot adequately separate the data, SVM can use a kernel function to transform the data into a higher-dimensional space, where it may become linearly separable. Common kernel functions include linear, polynomial, radial basis function (RBF), and sigmoid kernels.



**Linear Kernel:** K(xi​, x j​)=x i​⋅x j​. This is the identity function, preserving the original feature space.

**Polynomial Kernel:** polynomial kernel is one of the popular kernel functions used for mapping data into a higher-dimensional space in order to make it linearly separable.

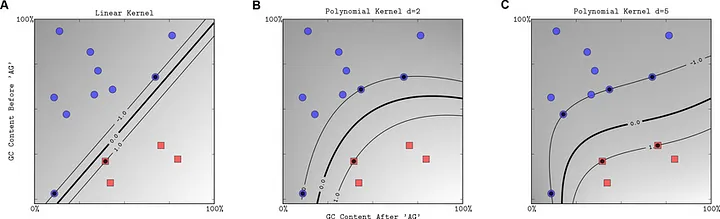
K(x, y) = (α \* (x ⋅ y) + c)^d

K(x, y) represents the kernel function for input vectors x and y.

α is a user-defined parameter that controls the influence of the dot product between x and y in the transformation.

c is a constant term, which can also be user-defined.

d is the degree of the polynomial. The degree of polynomial determines the complexity and flexibility of the model



**Radial Basis Function (RBF) Kernel:** The Radial Basis Function (RBF) kernel, also known as the Gaussian kernel, It is especially effective in handling non-linearly separable data by mapping it into a higher-dimensional space where linear separation is possible.

K(x i​,x j​)=exp(- ∥x i​−x j​∥ 2/ 2σ 2​),

where σ is a kernel parameter. The RBF kernel maps data into an infinite-dimensional space.

**Sigmoid Kernel:** The Sigmoid kernel is inspired by the sigmoid activation function used in neural networks. It transforms the input data into a higher-dimensional space, where it may or may not be linearly separable depending on the choice of parameters α and c.

K(x i​,x j​)=tanh(αx i​⋅x j​+β), where α and β are parameters. It maps data into a higher-dimensional space with a sigmoid-like transformation.

**C Parameter:** The C parameter controls the trade-off between maximizing the margin and minimizing classification errors.

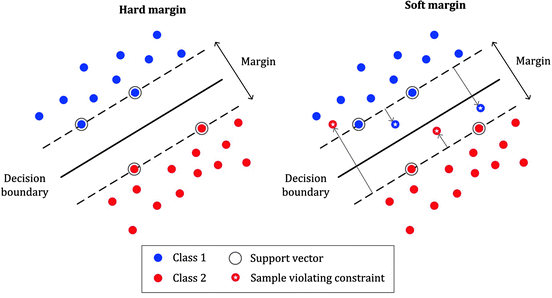
A smaller C value results in a larger margin but may allow some misclassification,

while a larger C value aims to minimize misclassification even if it reduces the margin.

**SVM with Soft Margin and Hard Margin:**

When C is very large (e.g., C = infinity), the SVM aims for a "hard margin" where it tries to perfectly classify all training data points, potentially leading to overfitting.

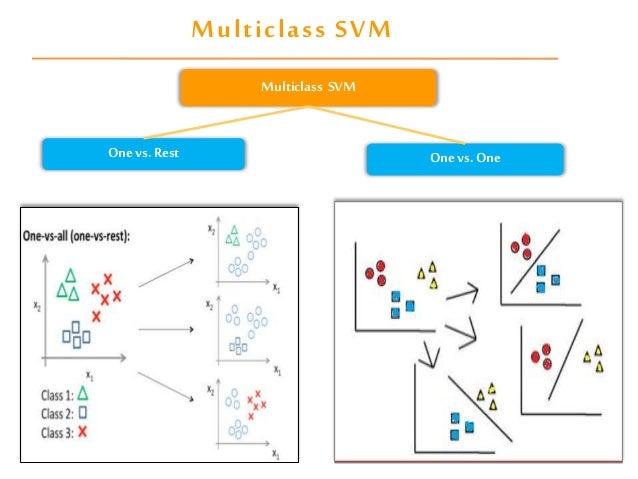
When C is small (e.g., C = 0), the SVM allows for a "soft margin" with a wider margin but may accept some misclassification.



**Multiclass Classification:** SVM can be extended to handle multiclass classification problems using methods like one-vs-one (OvO) or one-vs-all (OvA) strategies, where multiple binary classifiers are trained and combined to make multiclass predictions.

In the OvO strategy, SVM creates a binary classifier for each pair of classes in the multiclass problem. Suppose you have N classes. This strategy results in N(N-1)/2 binary classifiers.

In the OvA strategy, a single binary SVM classifier is trained for each class to distinguish that class from all the other classes combined.



Data Preprocessing: Prepare and preprocess the data, including feature scaling and encoding categorical variables.

Choose a Kernel: Select an appropriate kernel function based on the problem's characteristics.

Hyper parameter Tuning: Determine the optimal values for hyperparameters, such as C (regularization parameter) and kernel-specific parameters, using techniques like cross-validation.

Training: Train the SVM model on the training data, which involves finding the hyperplane that best separates the data.

Testing and Evaluation: Evaluate the model's performance on a separate test dataset using appropriate metrics (e.g., accuracy, F1-score, ROC curve).

Prediction: Use the trained SVM model to make predictions on new, unseen data.

Assumption of svm

* The main assumption of SVM is that the data is linearly separable. This means that there is a clear boundary between the different classes in the data.
* Another SVM is that the data is normalized. This means that the input features are scaled to have a mean of zero and a standard deviation of one.
* A third assumption of SVM is that the data is homogeneous.

Questions:

1. Linear SVM/Non Linear SVM
2. SVM kernel
3. Explain SVM
4. Explain SVM, compare RBF & Polynomial kernel
5. What is SVM with one example
6. What is SVM support vectors?
7. SVM(Polynomial)
8. What is kernel trick, explain its types?
9. What is sigmoid kernel?

10.What is kernel?