



Artificial Learning Models

Lecture 5 : Support Vector Machine

By : Dr. Lamri SAYAD

2024

Agenda

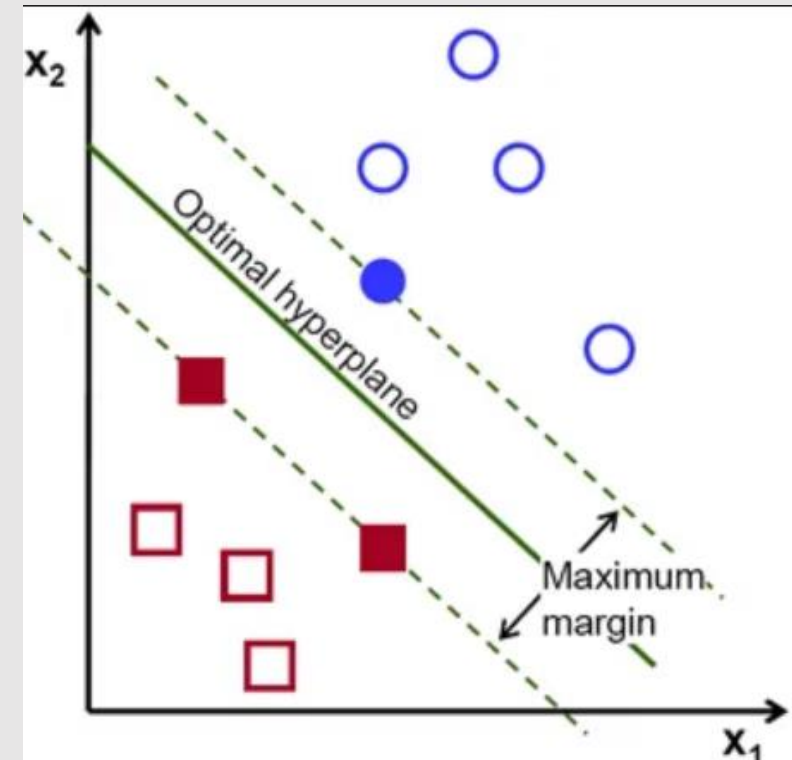
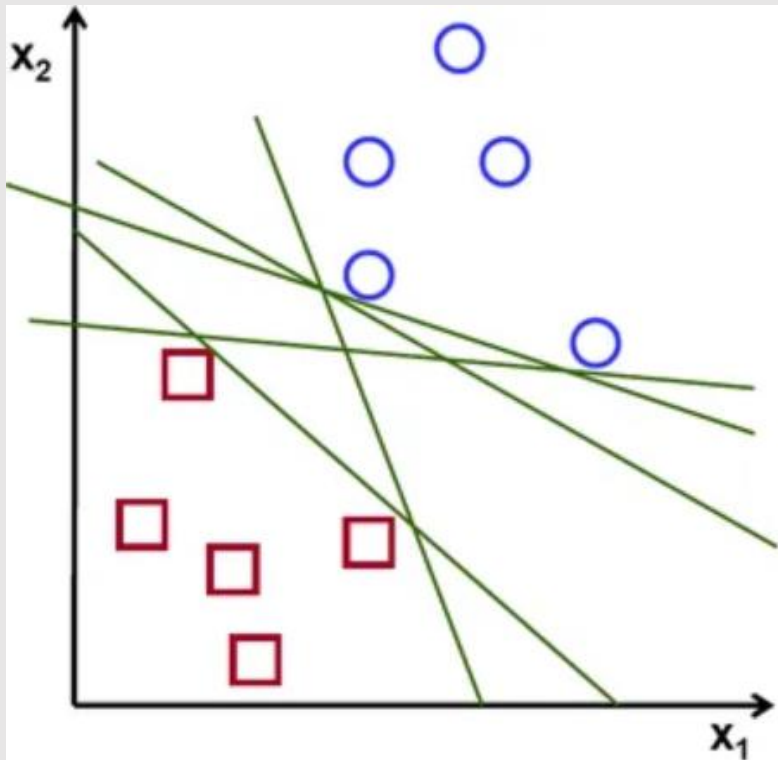
Introduction

- Supervised learning model
 - Used for classification
 - But also for regression
- Developed at **AT&T Bell Laboratories** by **Vladimir Vapnik** with colleagues (Boser et al., 1992, Guyon et al., 1993, Cortes and Vapnik, 1995, Vapnik et al., 1997)
- Is not a probabilistic classifier

What is SVM ?

- During training:
 - SVM constructs a hyperplane or set of hyperplanes in a high or infinite-dimensional space
 - SVM maps training examples to points in space so as to maximize the width of the gap between the classes.
- During test or prediction:
 - New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

What is SVM ?



How does SVM work ?

- It is defined in terms of the support vectors only,
 - Don't have to worry about other observations
- 
- The margin is made using the points which are closest to the hyperplane (support vectors)
-
- Remember
 - In logistic regression, the classifier is defined over all the points.

What is SVM ?

SVM vs Logistic regression

- Don't get confused
 - Both the algorithms try to find the best hyperplane
- The difference :
 - logistic regression is a probabilistic approach
 - SVM is based on statistical approaches.
- SVM or Logistic regression
 - SVM works best when the dataset is small and complex.
 - Use logistic regression first and see how does it perform, if it fails to give a good accuracy you can go for SVM without any kernel
 - Logistic regression and SVM without any kernel have similar performance

Types of SVM algorithms

- Linear SVM
 - The data is perfectly linearly separable
 - Performs linear classification
- Non-linear SVM
 - Data is not linearly separable
 - Perform a non-linear classification using some advanced techniques like kernel tricks to classify them.
 - kernel tricks ??!

SVM Terminology

- **Hyperplane**

- Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space.

- **Support vectors**

- These are the points that are closest to the hyperplane.
- A separating line will be defined with the help of these data points.

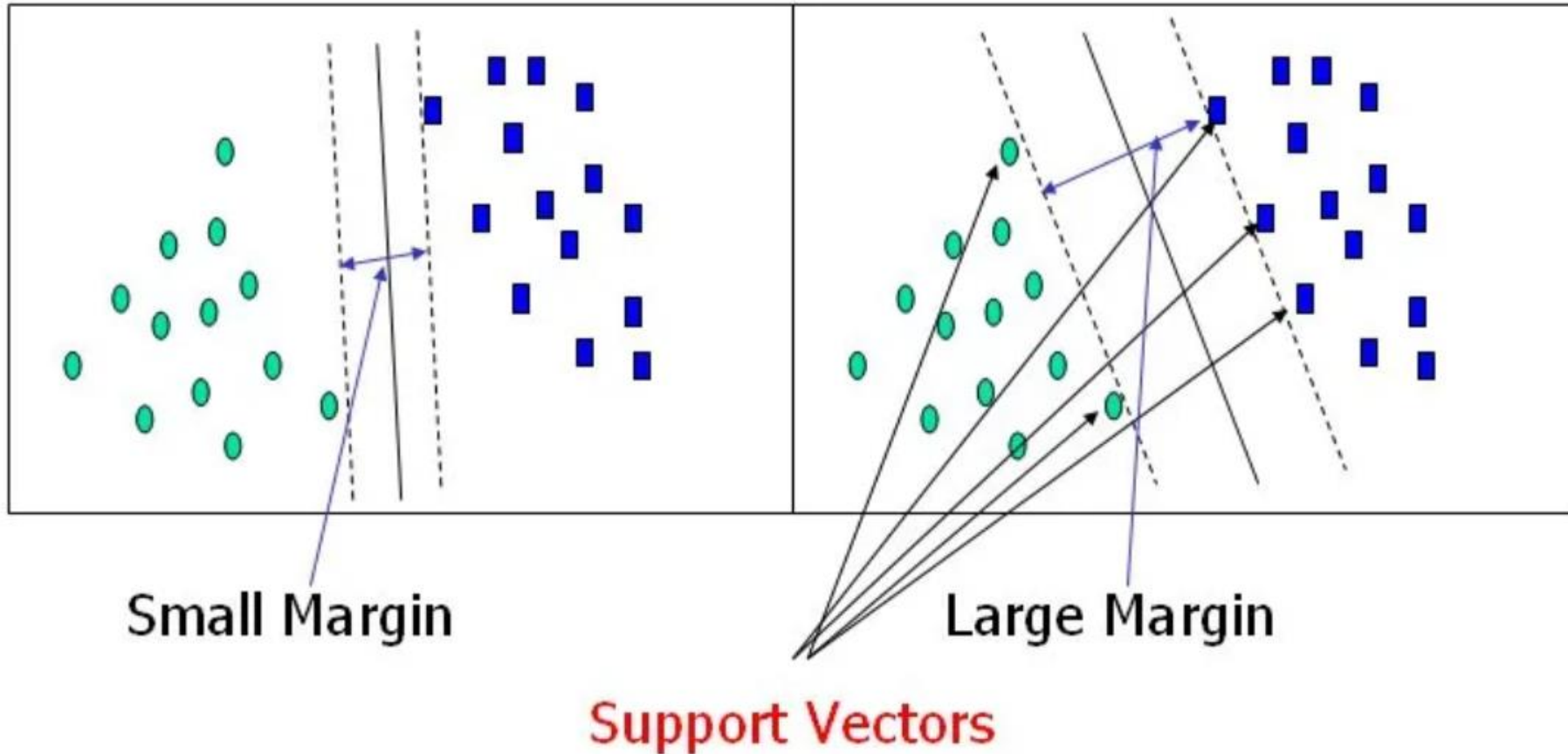
- **Margin**

- It is the distance between the hyperplane and the support vectors.
- Large margin is considered a good margin.
- Two types of margins **hard margin** and **soft margin**.

SVM Terminology

- **Kernel**
- **Hard Margin**
- **Soft Margin**
- **C**
- **Hinge Loss**

Support vectors and Margin

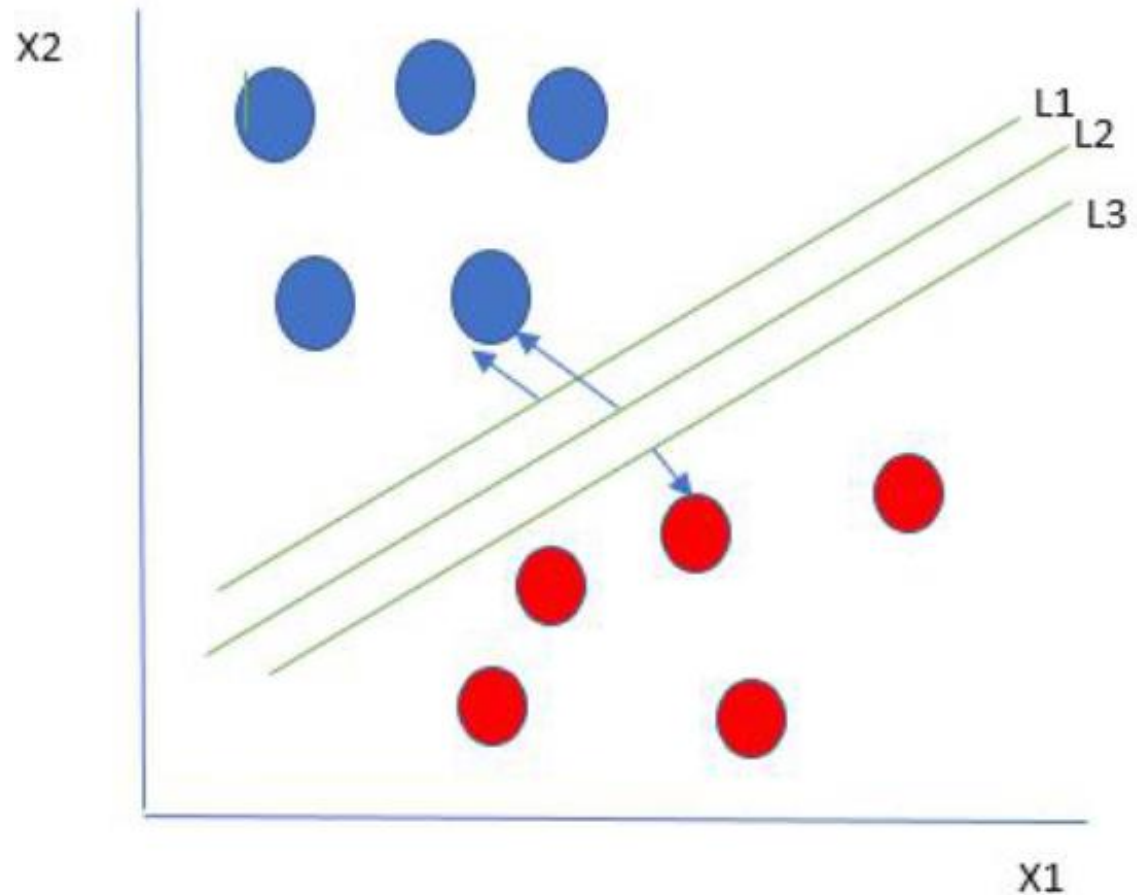


Hard margin

- Hyperplane with :
 - Perfect separation
 - Margin is maximized

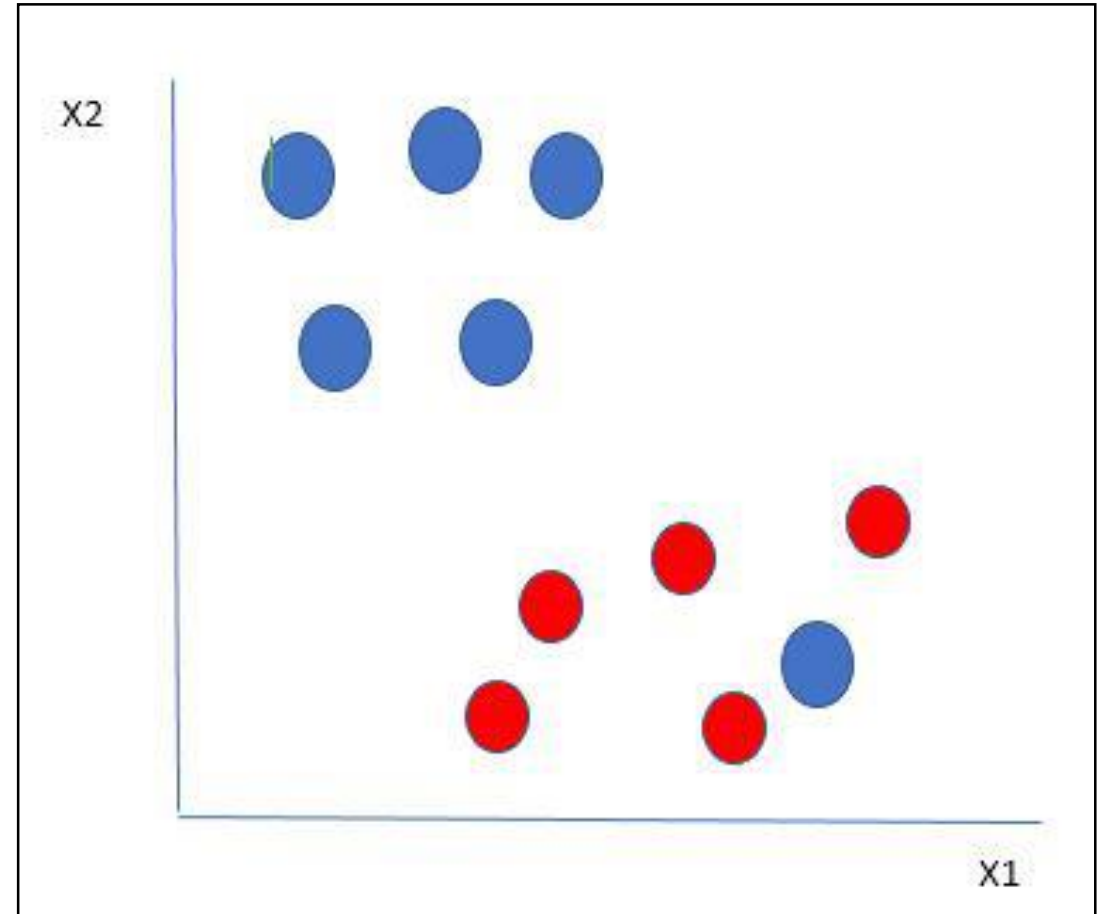


Hard Margin



Soft margin

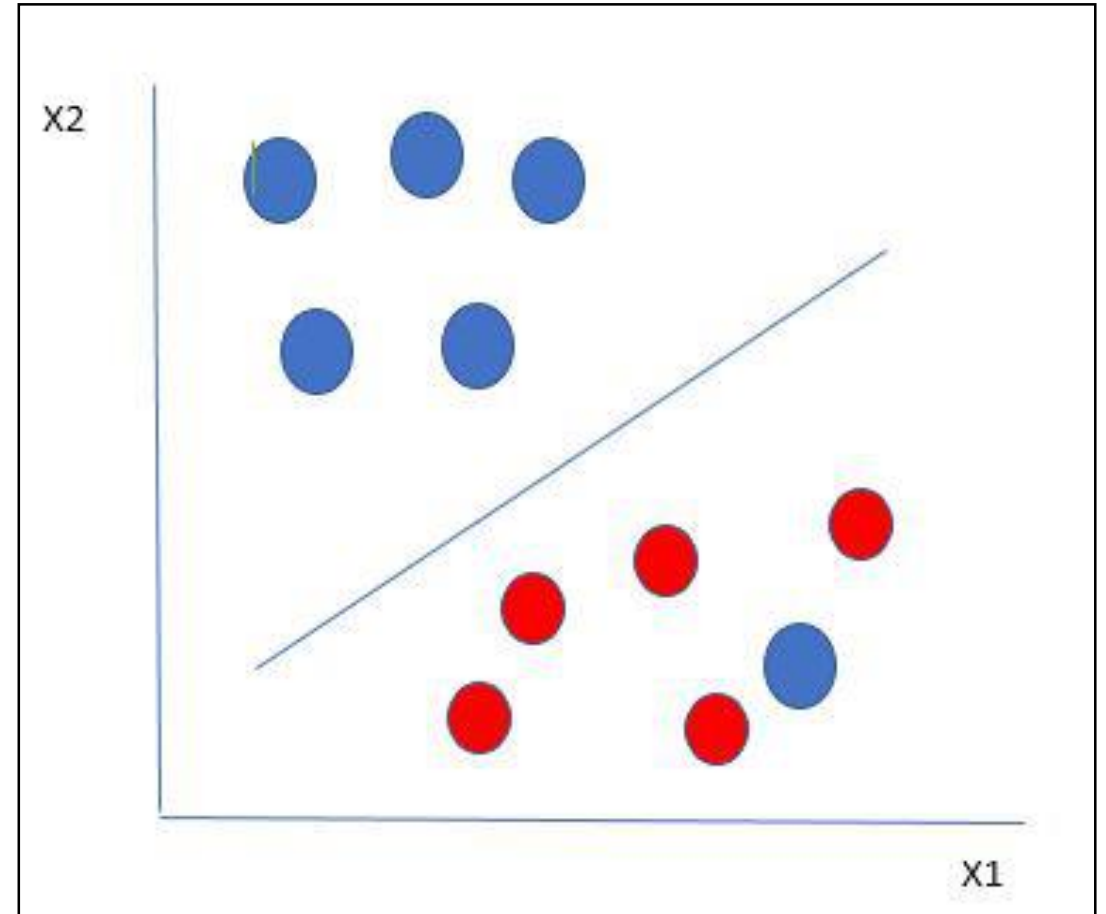
How does SVM classify these data?



Soft margin

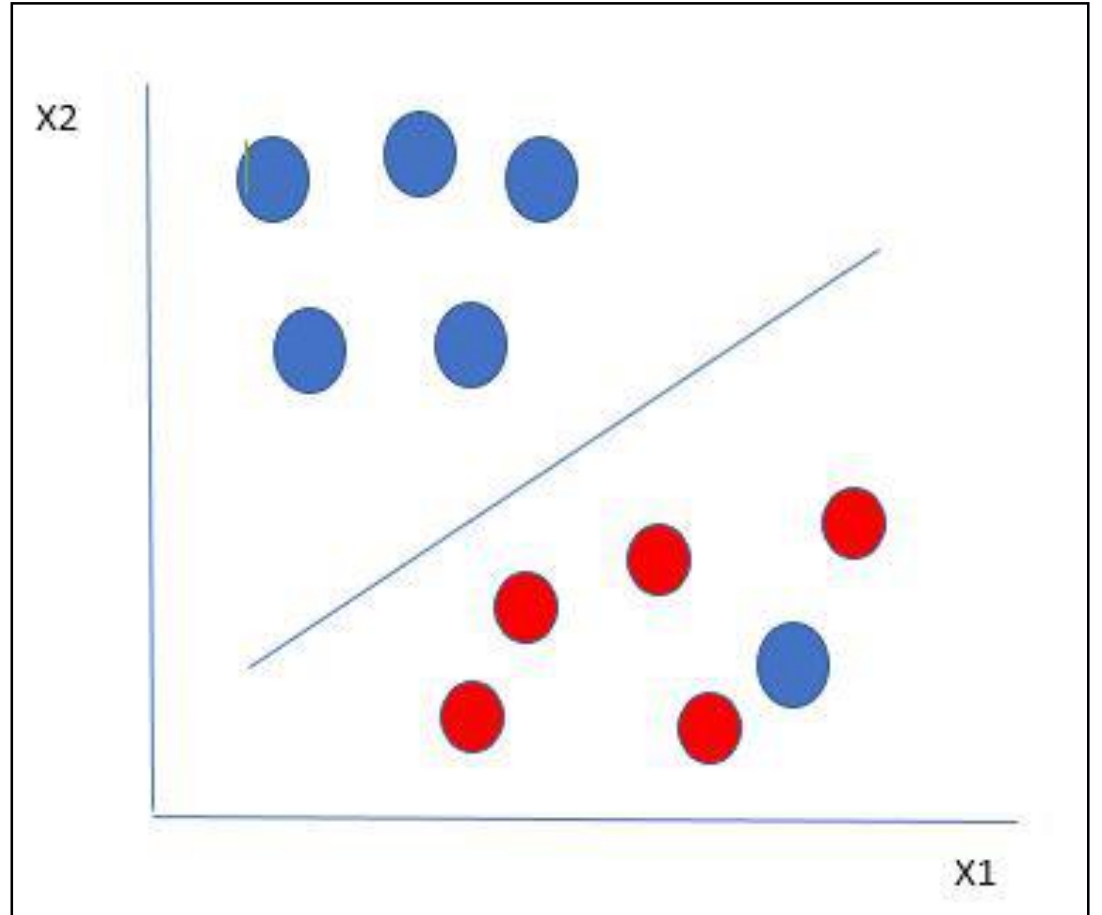
- The blue ball in the boundary of red ones is an outlier of blue balls.
- SVM ignore the outlier and finds the best hyperplane that maximizes the margin.
- SVM is robust to outliers.

How ??



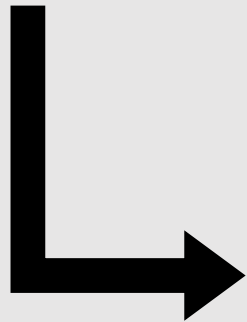
Soft margin

- Finds the maximum margin
- Adds a penalty each time a point crosses the margin.

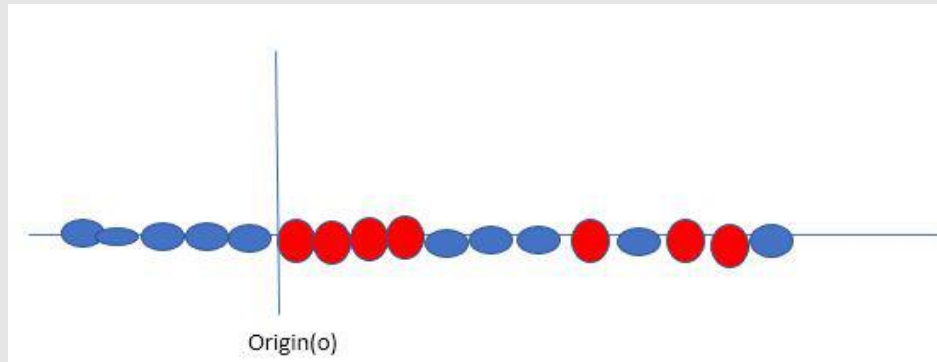


The kernel trick

- Till now, we were talking about linearly separable.



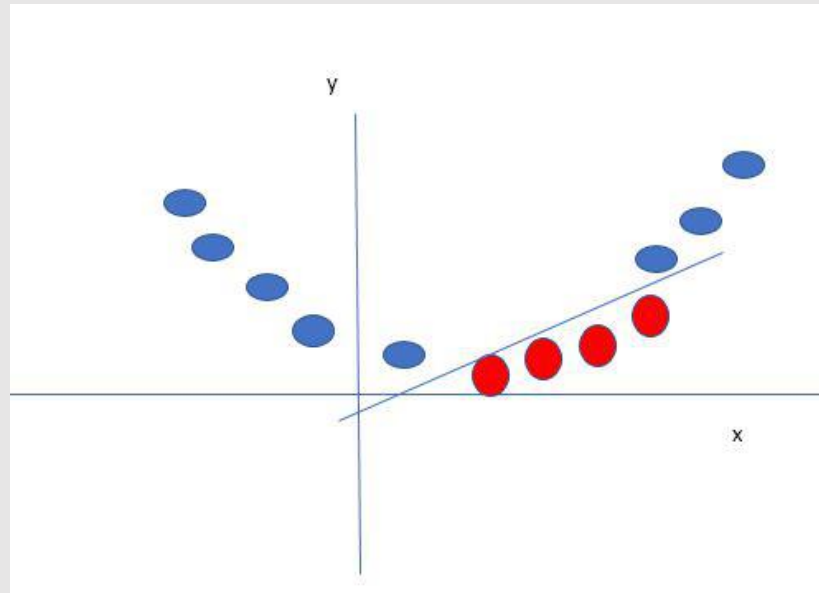
What to do if data are not linearly separable?



Create a new variable
using a **kernel**.

The kernel trick

- create a new variable y_i as a function of distance from origin o .
- A non-linear function that creates a new variable is referred to as a kernel.



The kernel trick

- Representing data in higher dimensional spaces to find hyperplanes that might not be apparent in lower dimensions

Kernel functions

- **Polynomial Kernel**

$$\text{Linear : } K(w, b) = w^T x + b$$

- **Sigmoid Kernel**

$$\text{Polynomial : } K(w, x) = (\gamma w^T x + b)^N$$

- **RBF Kernel**

$$\text{Gaussian RBF: } K(w, x) = \exp(-\gamma \|x_i - x_j\|^n)$$

$$\text{Sigmoid : } K(x_i, x_j) = \tanh(\alpha x_i^T x_j + b)$$

- **Bessel function kernel**

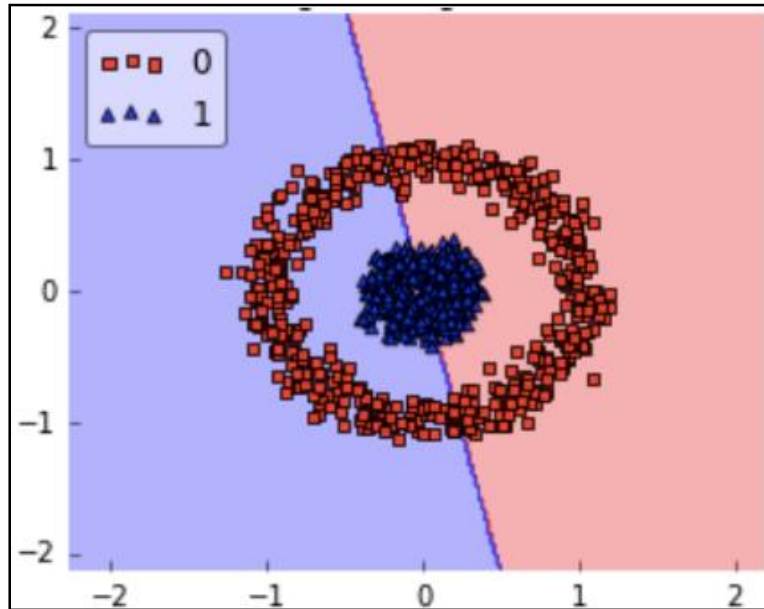
- **Anova Kernel**

How to choose the right kernel ?

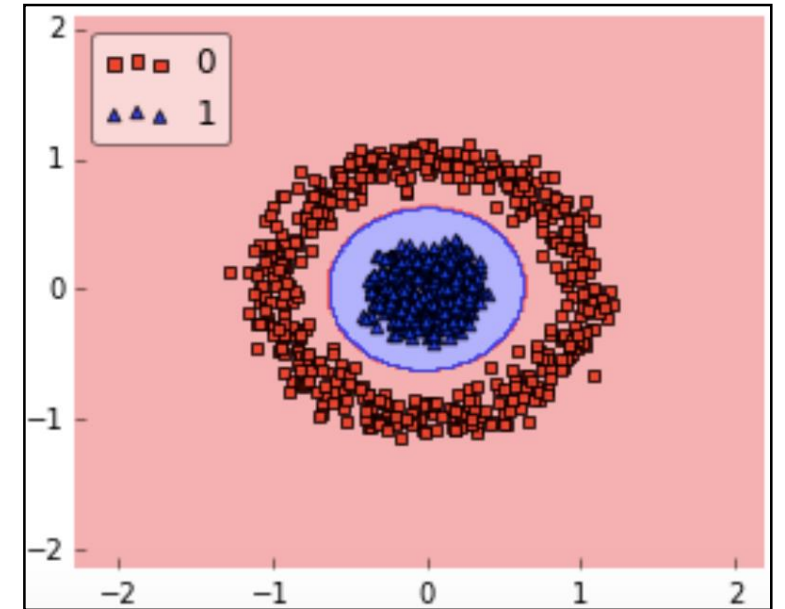
- The performance of the model depends on chosen kernel function
- Choosing a kernel totally depends on what kind of dataset are you working on
 - If it is linearly separable then you must opt for linear kernel function
 - It is recommended to start with a hypothesis that the data is linearly separable
- Usually, we use SVM with RBF and linear kernel function

How to choose the right kernel ?

Example



for this kind of dataset, we can use RBF without even a second thought because it makes decision boundary like this:



Mathematical intuition of SVM

- Consider a binary classification problem with two classes, labeled as +1 and -1.

- The equation for the linear hyperplane can be written as:

$$w^T x + b = 0$$

- The vector W represents the normal vector to the hyperplane. i.e the direction perpendicular to the hyperplane.
- The distance between a data point x_i and the decision boundary can be calculated as:

$$d_i = \frac{w^T x_i + b}{||w||}$$

- where $||w||$ represents the Euclidean norm of the weight vector w .

Mathematical intuition of SVM

Optimization

- For Hard margin linear SVM classifier:

$$\begin{aligned} & \underset{w,b}{\text{minimize}} \frac{1}{2} w^T w = \underset{W,b}{\text{minimize}} \frac{1}{2} \|w\|^2 \\ & \text{subject to } y_i(w^T x_i + b) \geq 1 \text{ for } i = 1, 2, 3, \dots, m \end{aligned}$$

- For Soft margin linear SVM classifier:

$$\begin{aligned} & \underset{w,b}{\text{minimize}} \frac{1}{2} w^T w + C \sum_{i=1}^m \zeta_i \\ & \text{subject to } y_i(w^T x_i + b) \geq 1 - \zeta_i \text{ and } \zeta_i \geq 0 \text{ for } i = 1, 2, 3, \dots, m \end{aligned}$$

Advantages and Disadvantages of SVM

- **Advantages of SVM**

- SVM works better when the data is Linear
- It is more effective in high dimensions
- SVMs are less prone to overfitting than other algorithms such as neural networks.
- SVM is not sensitive to outliers
- With the help of the kernel trick, we can solve any complex problem
- Can help us with Image classification

- **Disadvantages of SVM**

- Choosing a good kernel is not easy
- It doesn't show good results on a big dataset
- The SVM hyperparameters are Cost -C and gamma. It is not that easy to fine-tune these hyper-parameters. It is hard to visualize their impact

SVM with sklearn

- **Library** : *from sklearn import svm*
- **define the model** : `model = svm.SVC(kernel='linear', C=1.0)`
- **train the model** : `model.fit(training_X, training_y)`
- **make non-linear algorithm** : for model : `nonlinear_model = svm.SVC(kernel='rbf', C=1.0)`