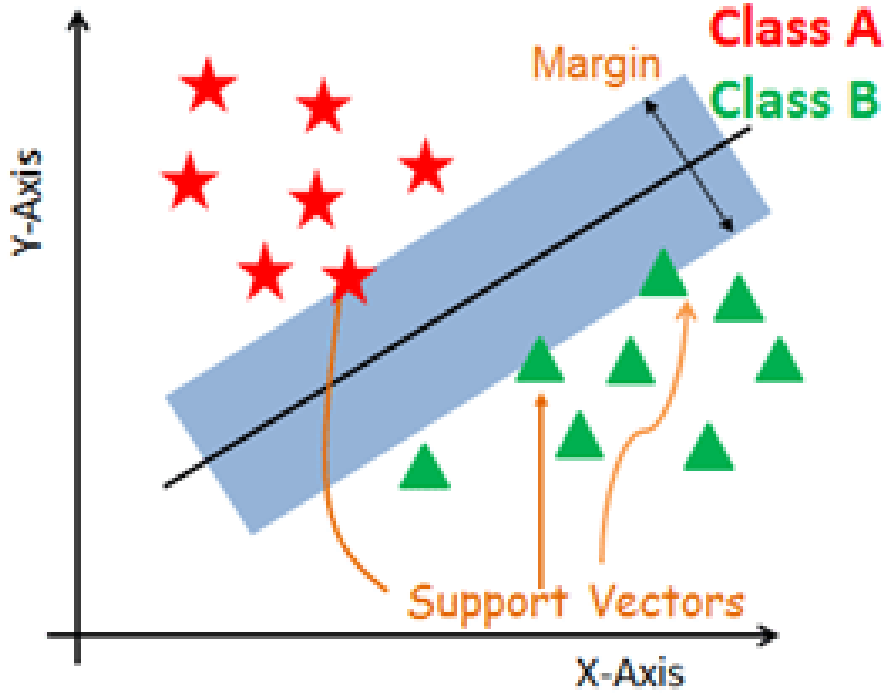
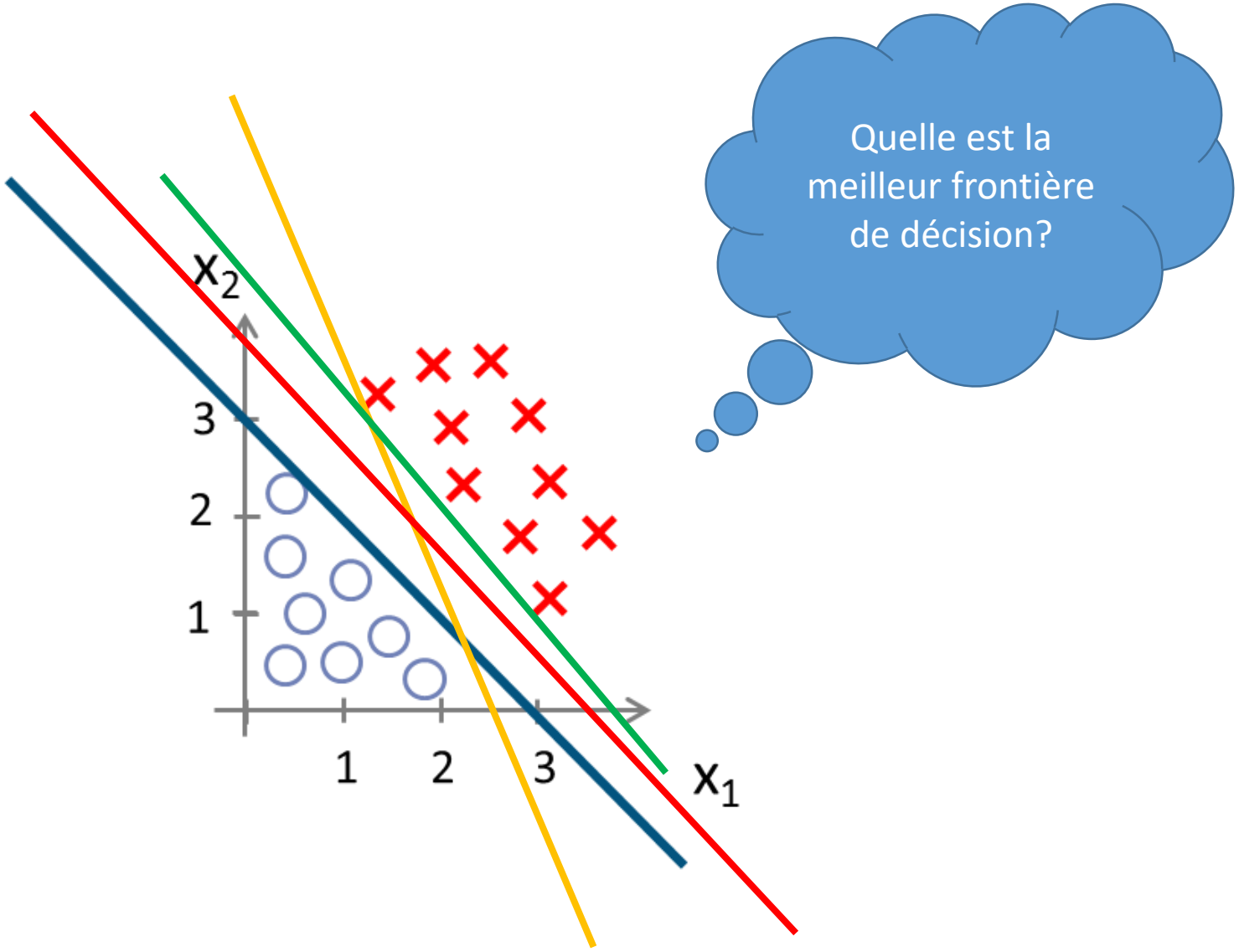
Plan détaillé du cours

1. Grande Vision d'Ensemble
2. Algorithmes célèbres de ML
   - Machines à Vecteurs de support (Support Vector Machines SVM)
   - Arbres de décision (Decision Trees)
   - Forêts aléatoires (Random Forest)
   - Autres algorithmes
   - Quel algorithme choisir ?

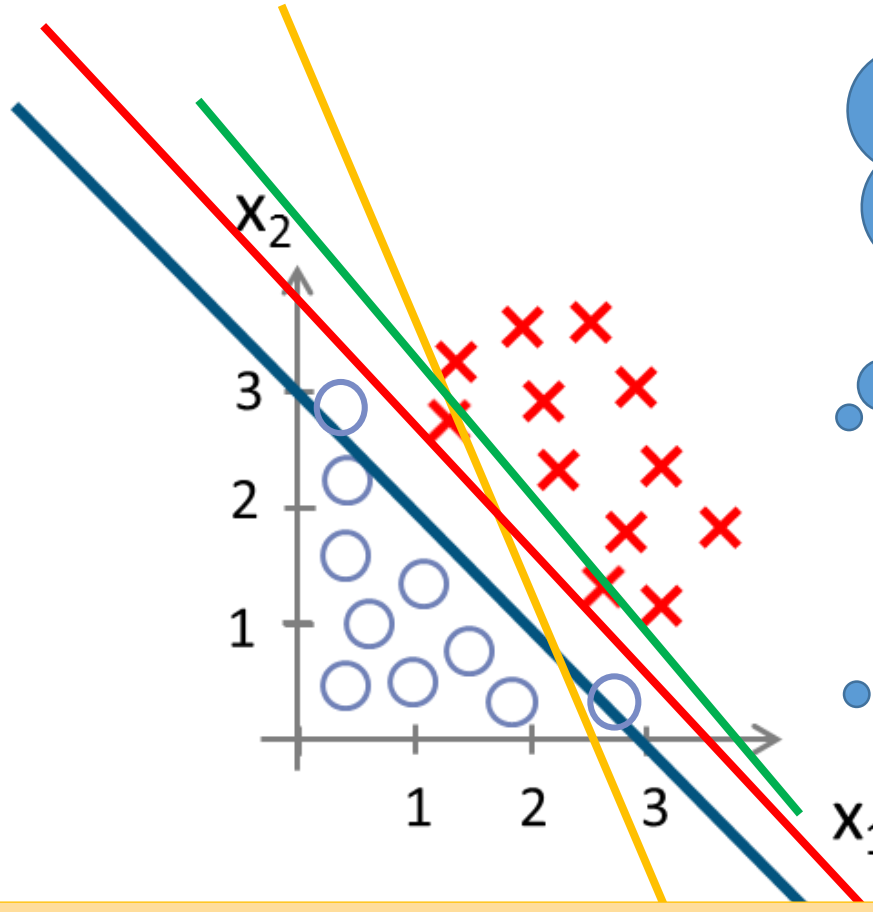
SVM

Support Vector machines




Quelle est la meilleur frontière de décision?
x_2
3
2
1
1
2
3
x_1

Si on rajoute des exemples …

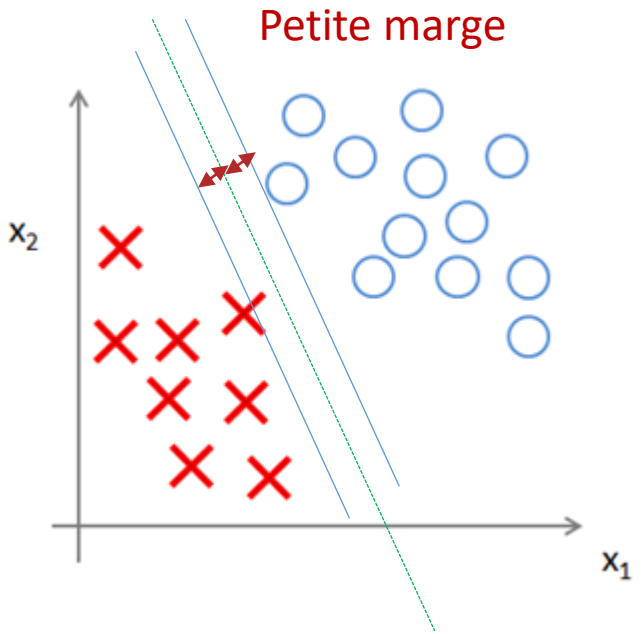


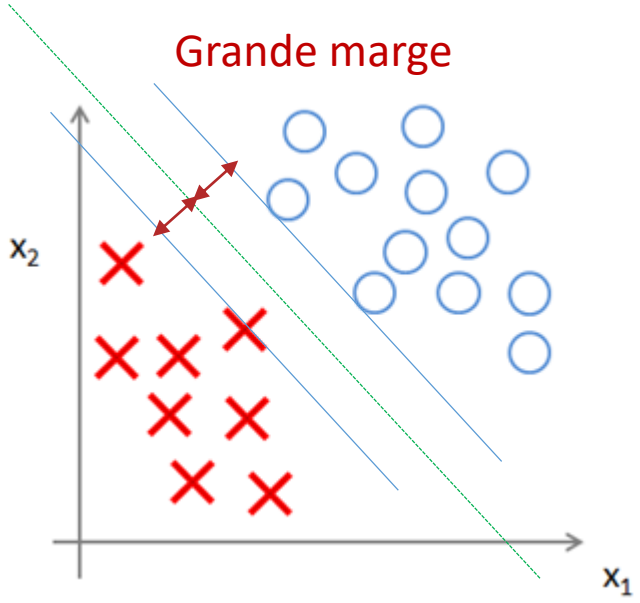
Quelle est la meilleur frontière de décision?

Qu'est ce qui fait qu'une frontière de décision est la meilleure??

Remarquons que:

- une **mauvaise frontière** de décision induit une mauvaise **généralisation**.
- La meilleur droite est celle qui est la plus éloignée des exemples





Remarquons que:
La frontière de décision la **plus éloignés possible** des exemples **positifs** et **négatifs** est celle avec **la plus grande marge entre les deux classes.**

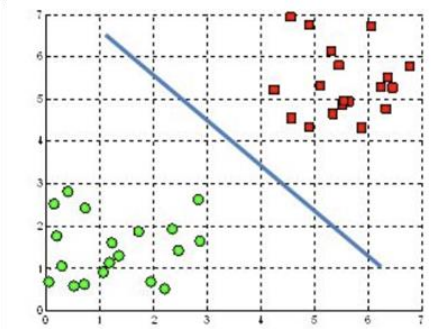
Support Vector Machines (SVM)

Un des plus **performants** et des plus **populaires** algorithms de **Classification!**

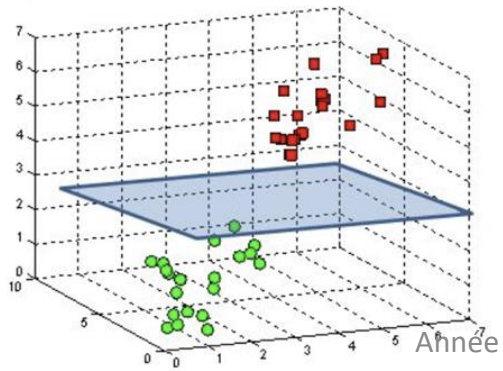
Comme pour la Régression Logistique, SVM construit un **hyperplan** qui sépare la classe **positive** de la classe **négative**

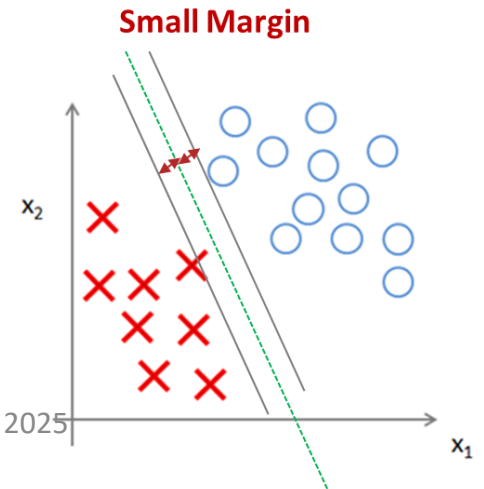
La frontière de décision de SVM a la particularité d'être la **plus éloignés possible** des exemples **positifs** et **négatifs:**
Maximum de marge entre les deux classes

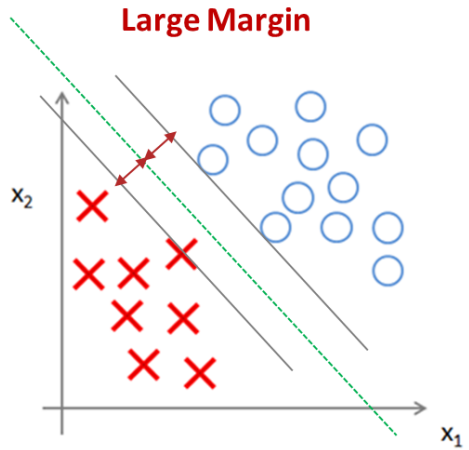
2-dimensions:
Hyperplan = une ligne



3-dimensions:
Hyperplan = un plan





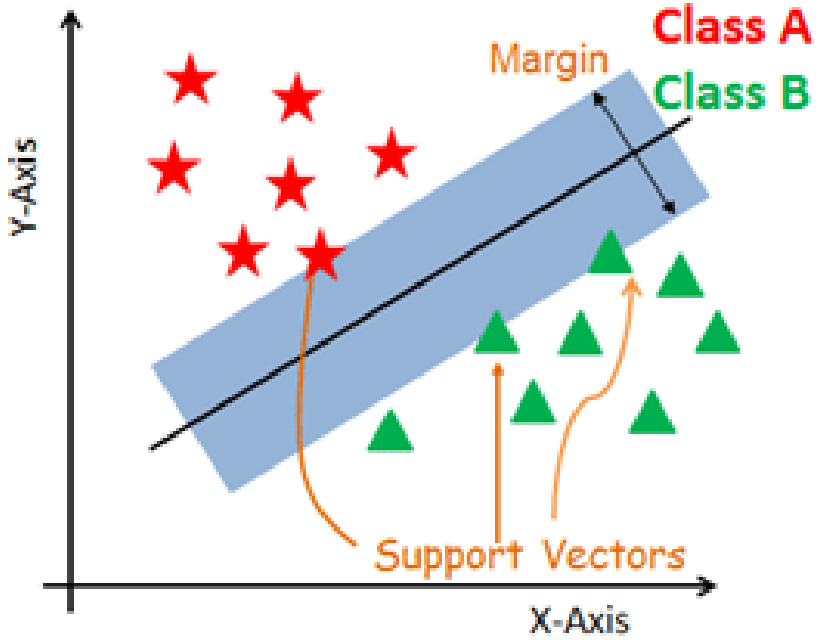


Support Vector Machines (SVM)

- Un des **plus performants** et des **plus populaires** algorithmes de **classification.**

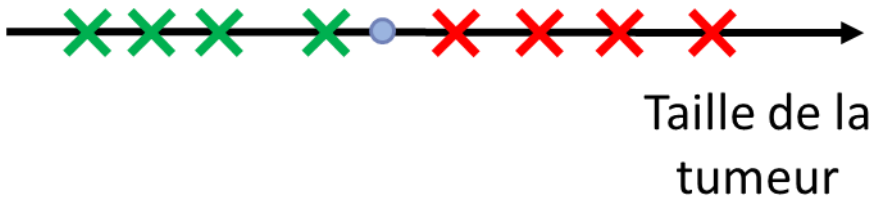
Principe de Fonctionnement de SVM:

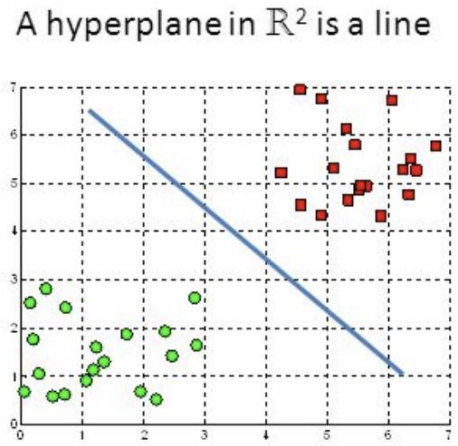
1. Construit un **hyperplan** qui sépare les deux classes
2. **Maximum de marge** entre les deux classes
3. S'appuie sur certain vecteurs (**support vectors**)
4. Paramètre **C**
5. **Frontières non linéaires grâce au noyaux (kernels)**



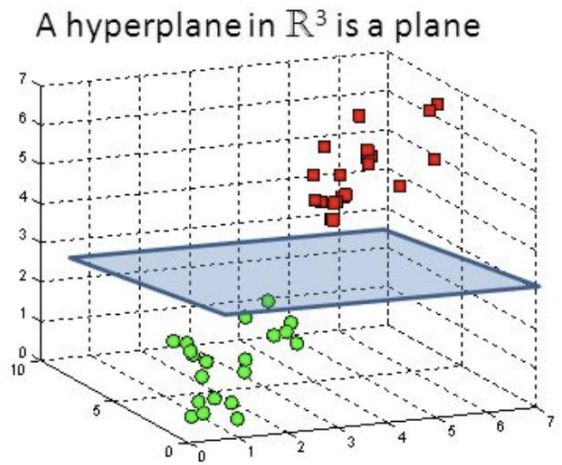
(1) Frontière de Décision en Hyperplan

1-dimension:
Hyperplan = un point





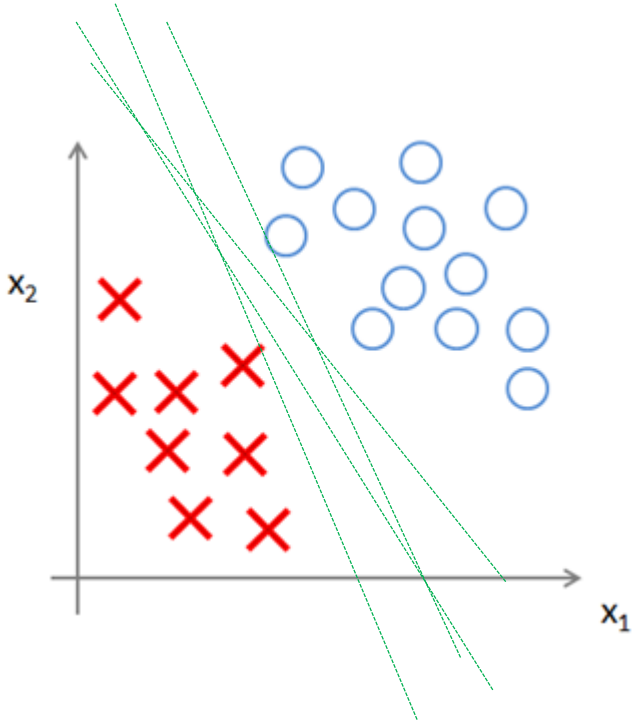
2-dimensions:
Hyperplan = une ligne

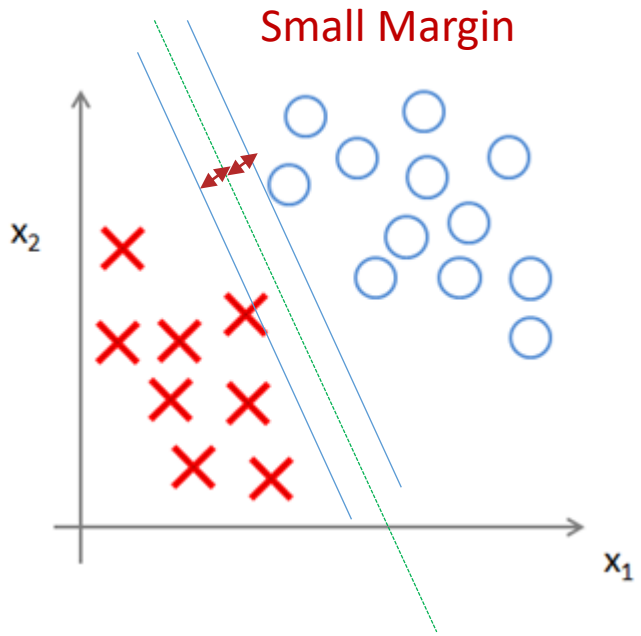


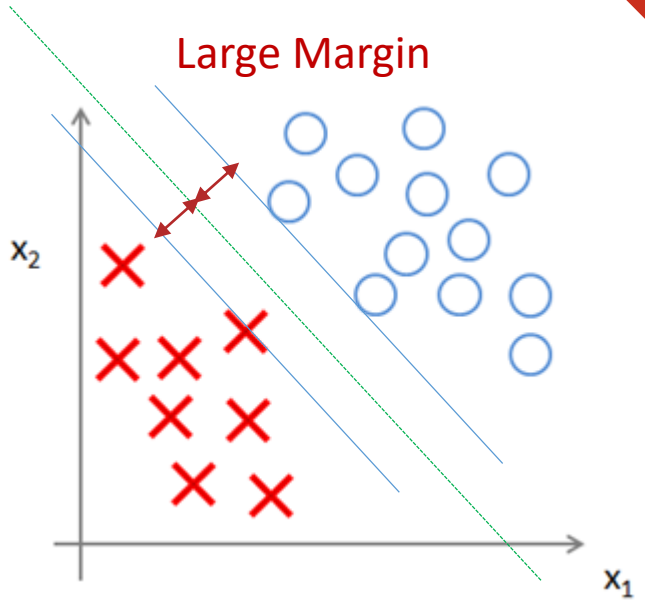
3-dimensions:
Hyperplan = un plan

Comme pour la Régression Logistique, SVM construit un **hyperplan** qui sépare la classe **positive** de la classe **négative**

SVM – (2) Maximum de Marge

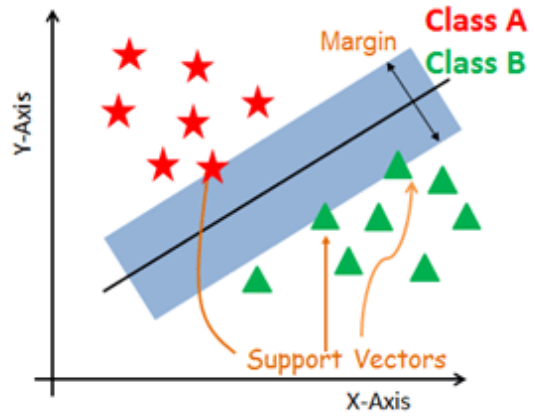


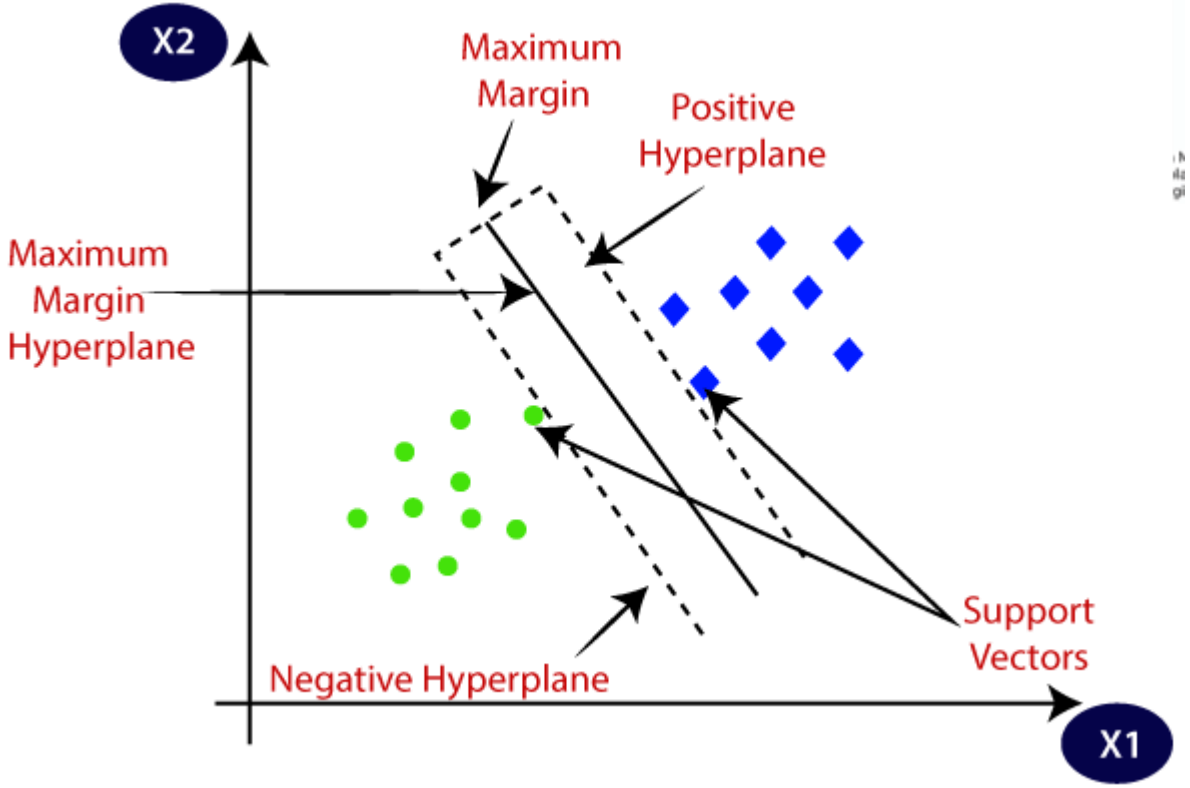


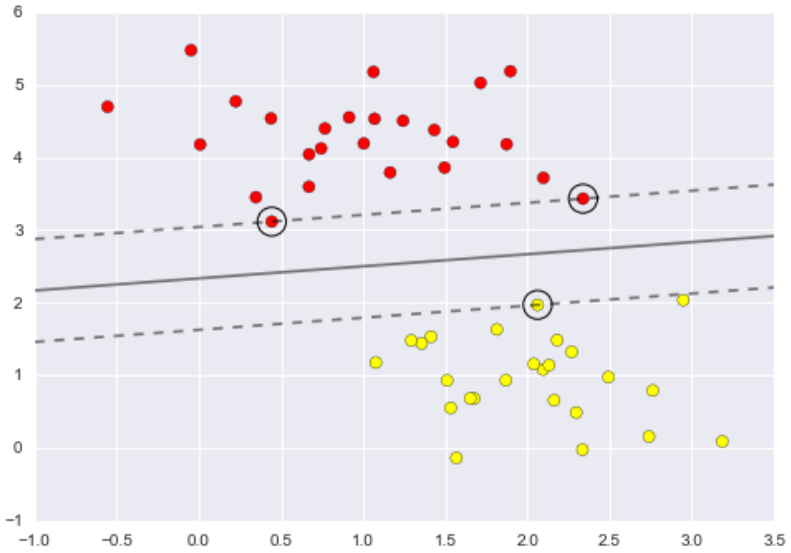


La frontière de décision de SVM a la particularité d'être la **plus éloignés possible** des exemples **positifs** et **négatifs**

(3) Vecteurs de support (Support Vectors)

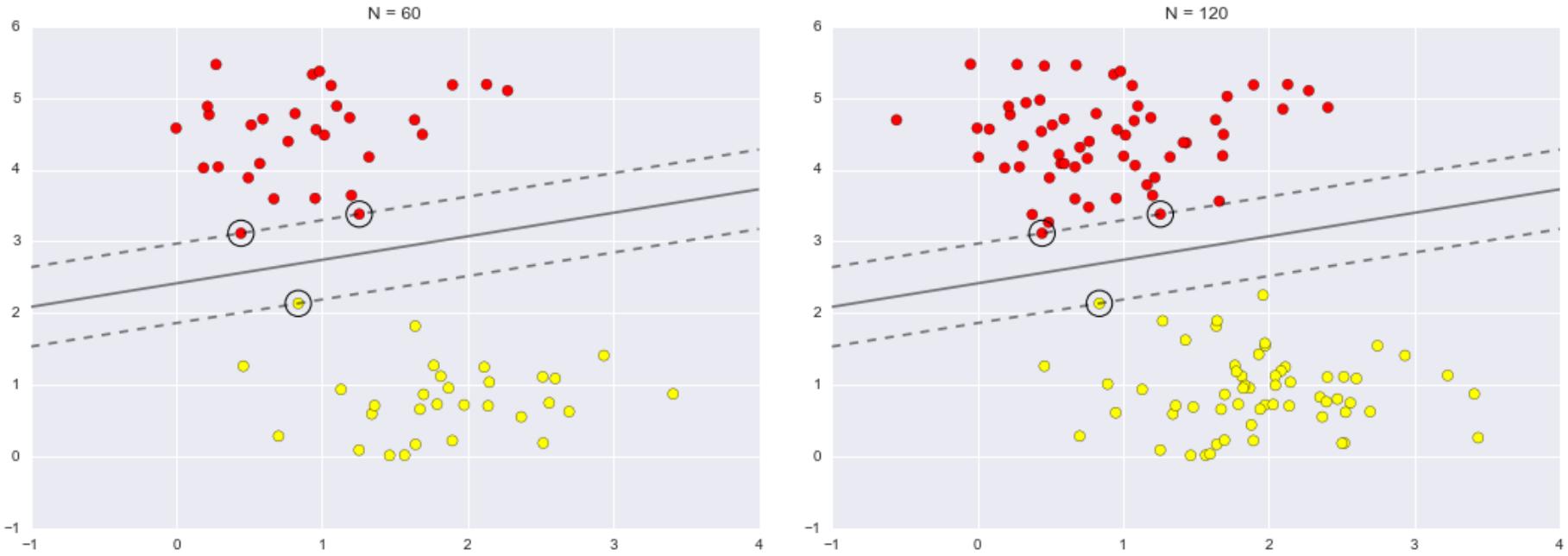






Vecteurs de support: les points les plus proches de l'hyperplan et qui influencent sa position et son orientation.

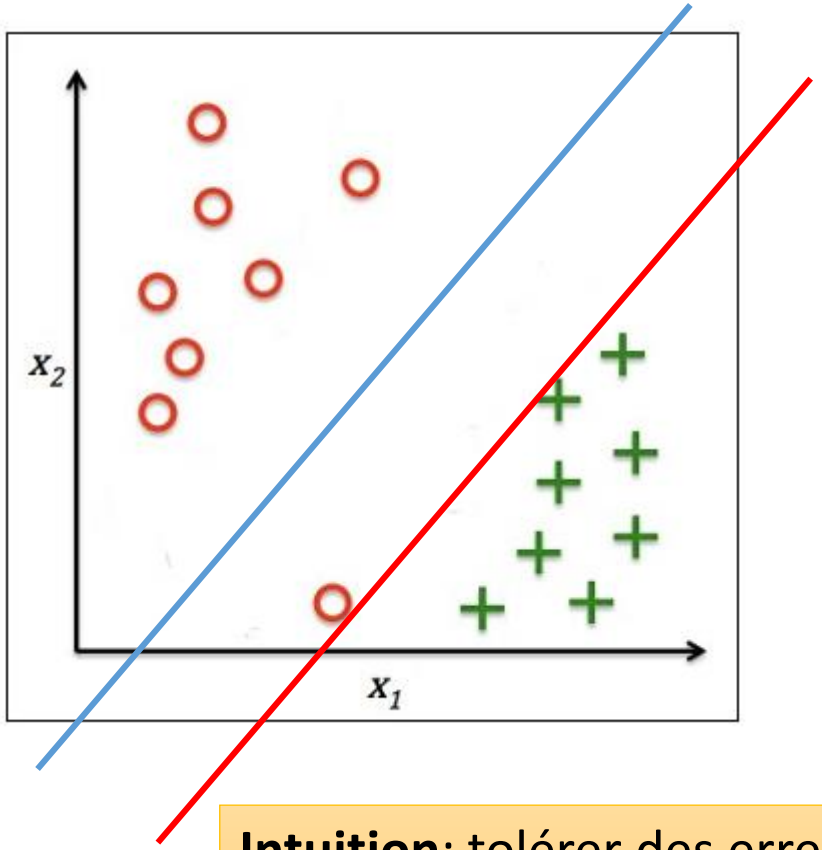
(3) Vecteurs de support (Support Vectors)



- **Uniquement** la position des vecteurs de support affecte la frontière de décision.
- Tout les autres points (et qui sont dans le bon côté) n'affectent pas la frontière de décision
- Une autre **force** de **SVM**: La **non-sensitivité aux points distants**!

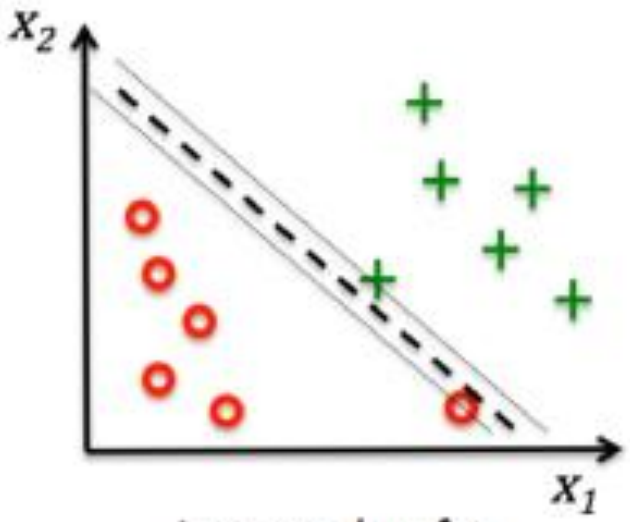
(4) Paramètre C

- Quelle est la meilleur frontière de décision?

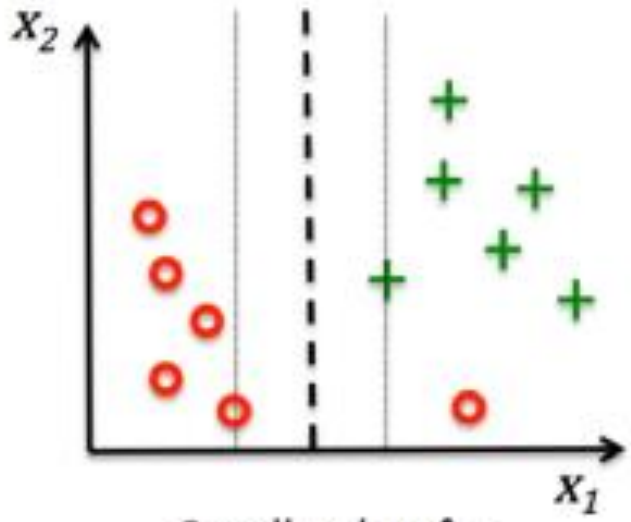


Intuition: tolérer des erreurs de classification pour une meilleure frontière de décision

(4) Paramètre C



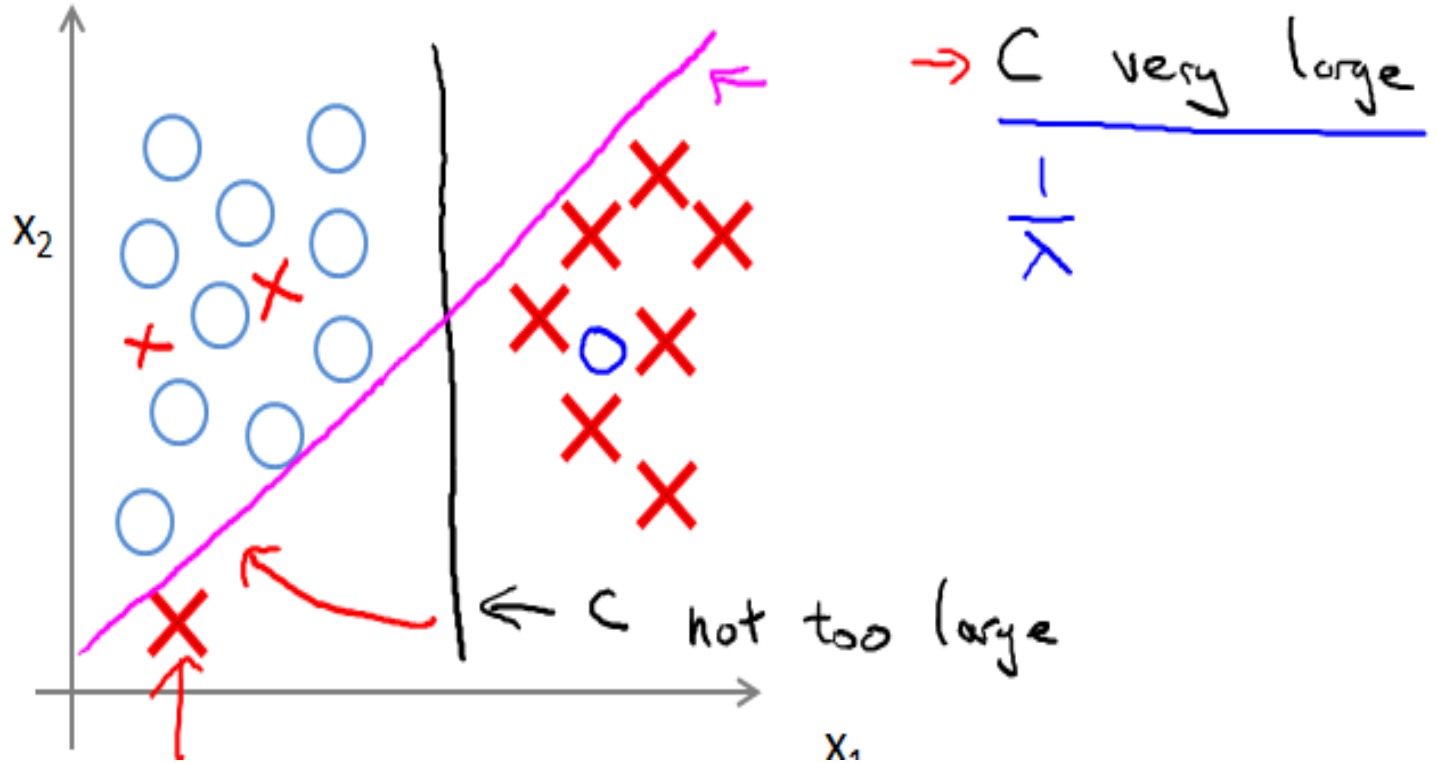
Large value for parameter *C*



Small value for parameter *C*

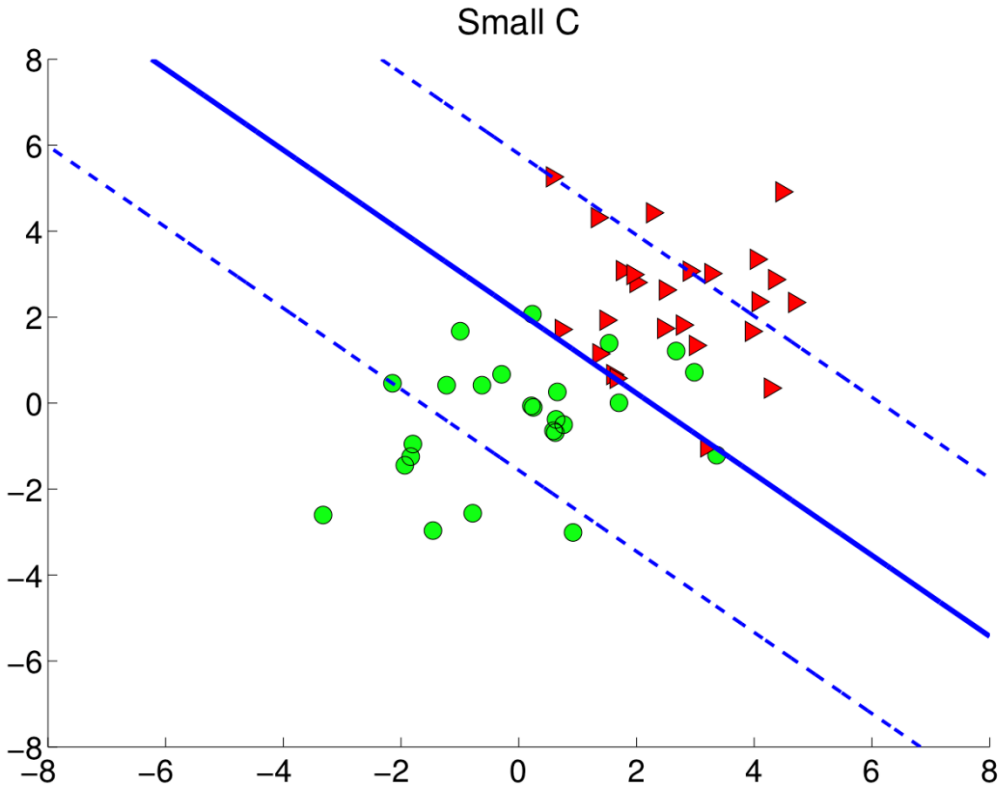
- **Paramètre C** : Permet de considérer ou non la '**misclassification**'.
- Equivalent à 1/λ (λ est la force de regularisation)
- Un **petit C** permet de tolérer des erreurs de classifications
- Un **grand C** respecte la contrainte de marge maximale mais avec zero erreurs.
- Un **petit C** viole la contrainte de marge maximale.

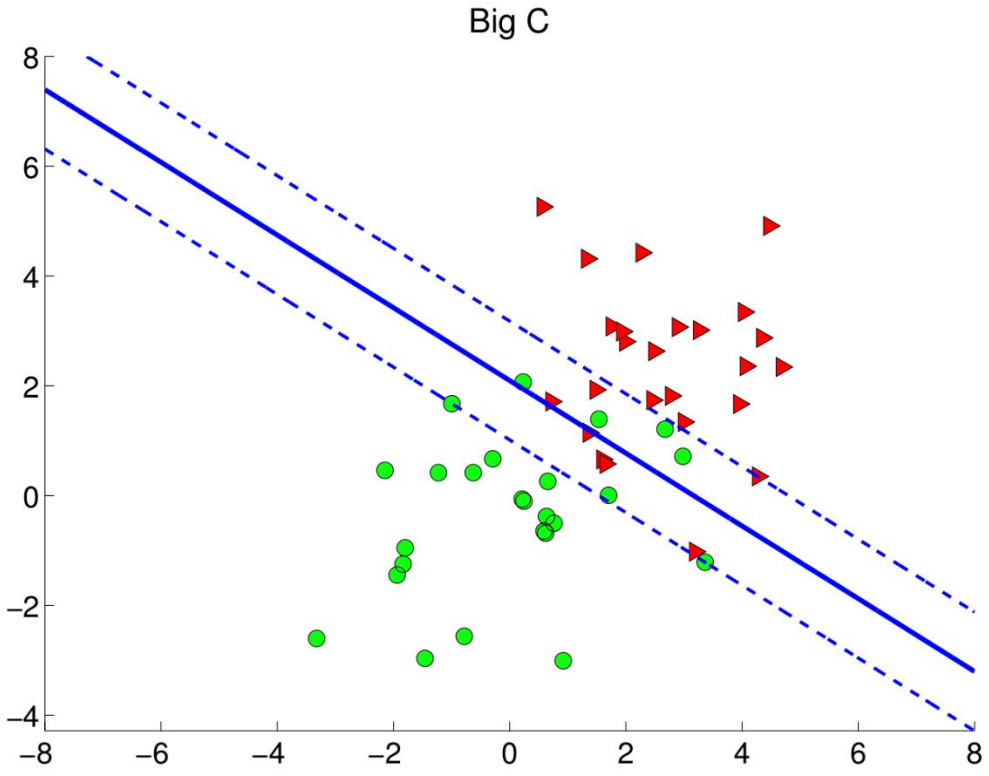
(4) Paramètre C



S'il y a présence de valeurs aberrantes qu'on désire ne pas affecter la frontière de décision, il faut alors **réduire C.**

(4) Paramètre C





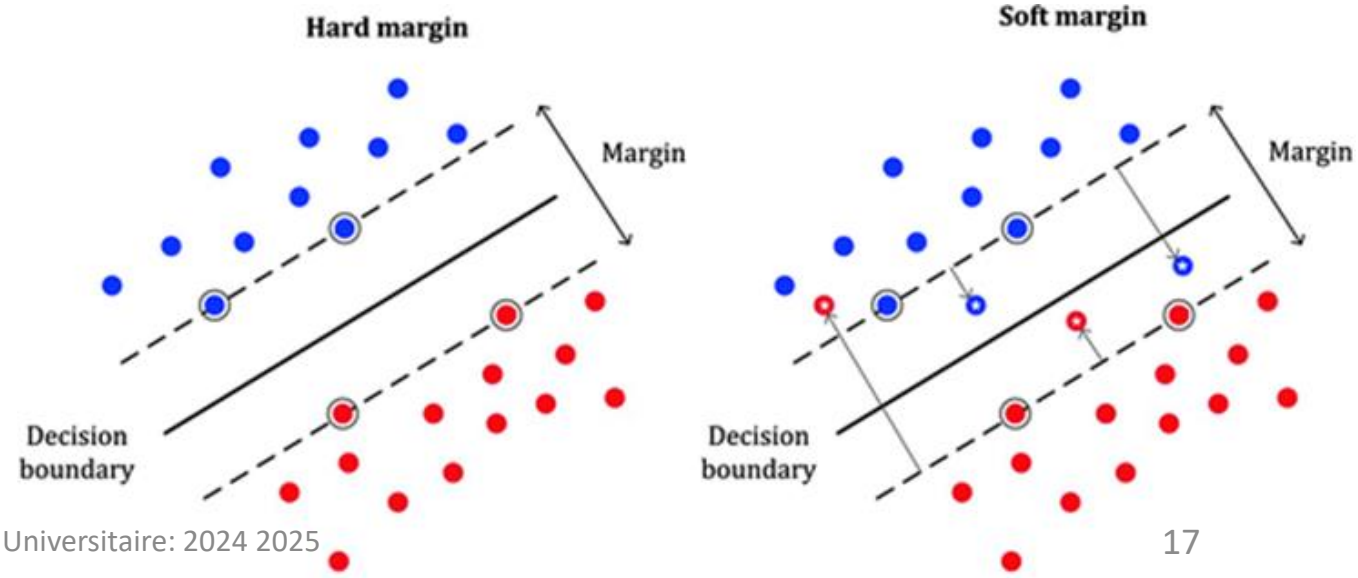
(4) Paramètre C

SVM à marge dure (Hard Margin) :

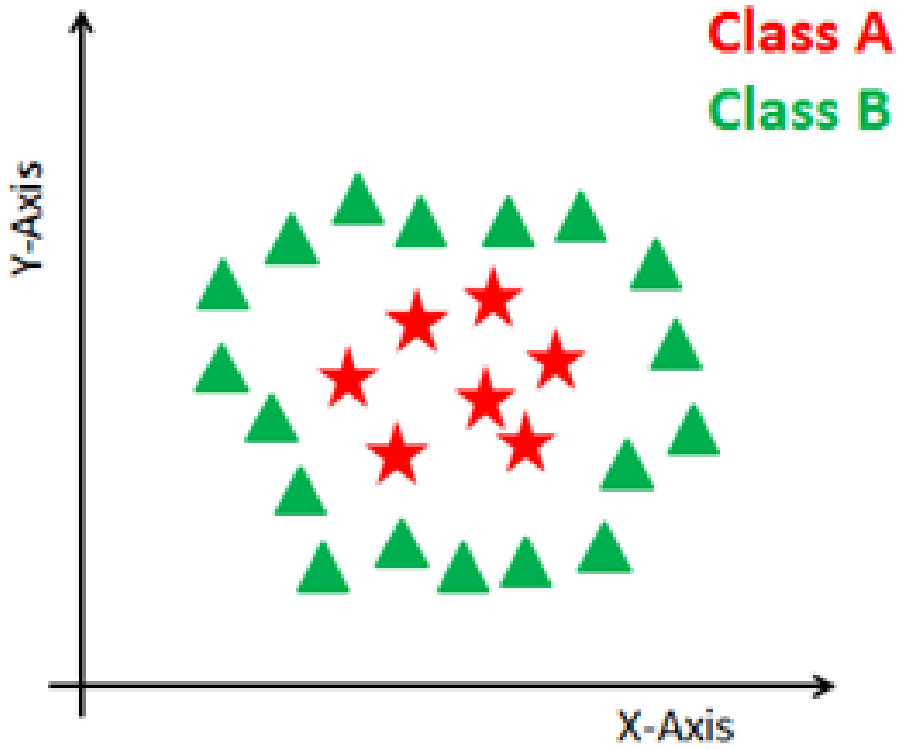
- chaque point de données doit être classé correctement.
- Plus sensible aux données bruitées et aux valeurs aberrantes.
- Peut ne pas généraliser correctement.

SVM à marge souple (Soft Margin) :

- peut tolérer un certain nombre d'erreurs de classification.
- Plus robuste aux données bruitées et aux valeurs aberrantes.
- Meilleur généralisation.



(5) Noyaux de SVM

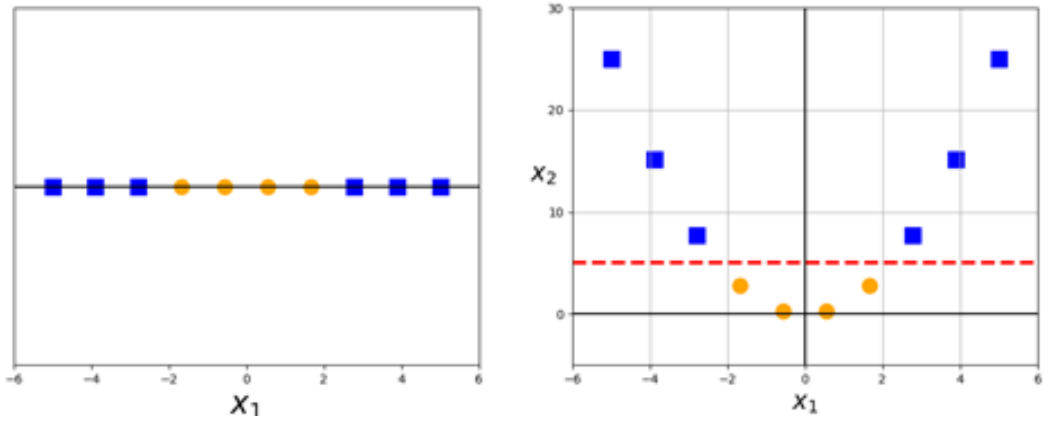


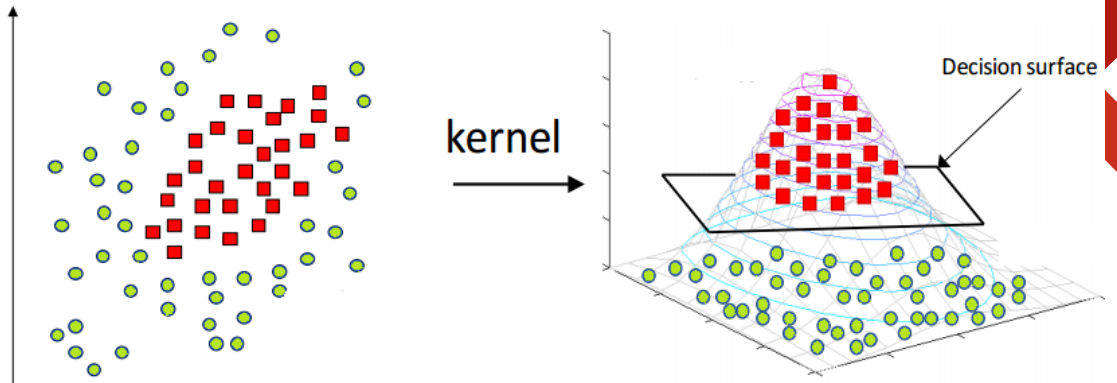
Données non-linéairement séparables:
Comment tracer **l'hyperplan** frontière de décision?

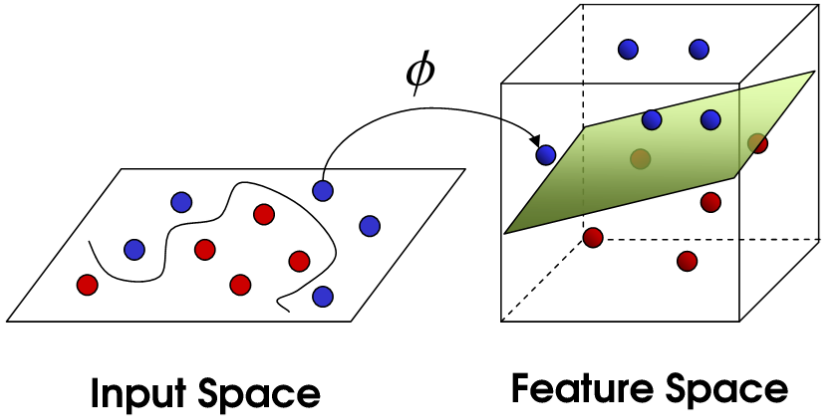
?

(5) Noyaux de SVM

On rajoute une nouvelle dimension $x_2 = x^2$





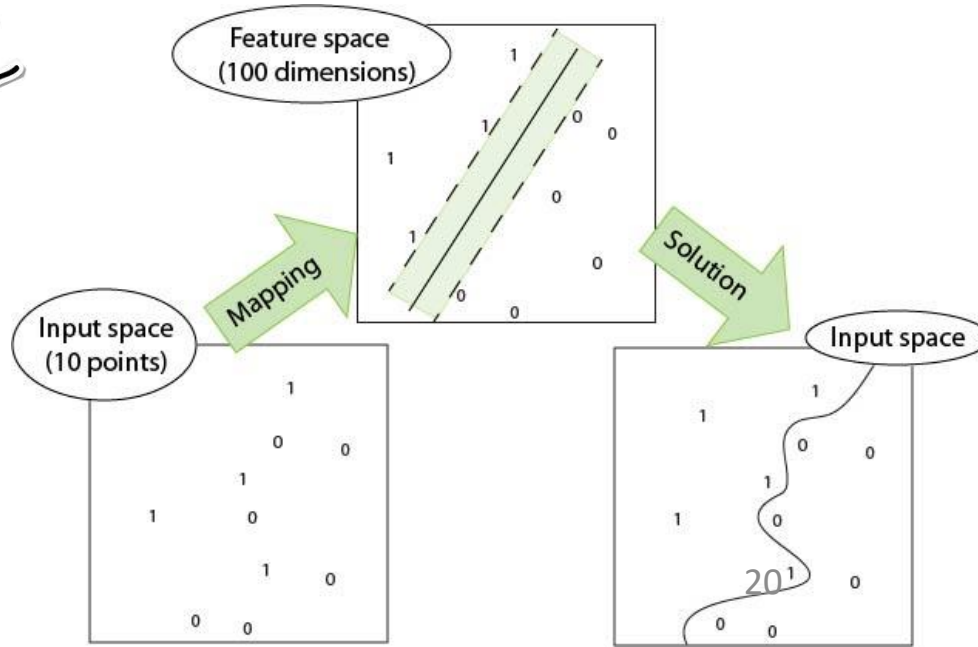


Objectif: transformer les données dans un **nouvel espace** (dit **espace caractéristique étendu**) de sorte à ce qu'elles deviennent **linéairement séparables** !

(5) Noyaux de SVM - Astuce du Noyau (kernel Trick)

- **Problème:** La transformation de données dans de nouvelles dimensions peut être très couteuse en calculs, surtout quand le nombre de caractéristiques est élevé.

- **Solution:** **Astuce du Noyau (Kernel trick)**



(5) Astuce du Noyau

- Un noyau (kernel) est une fonction qui calcule le produit scalaire entre deux vecteurs dans l'espace de caractéristiques étendu.

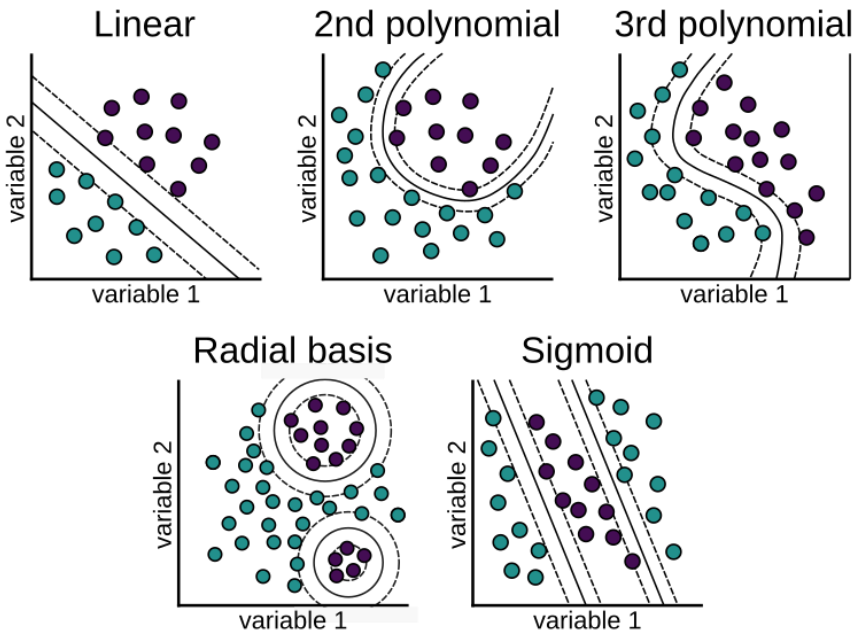
- **Astuce du noyau de SVM:** consiste à calculer les produits scalaires dans l'espace étendu sans avoir à effectuer explicitement les transformations vers cet espace.

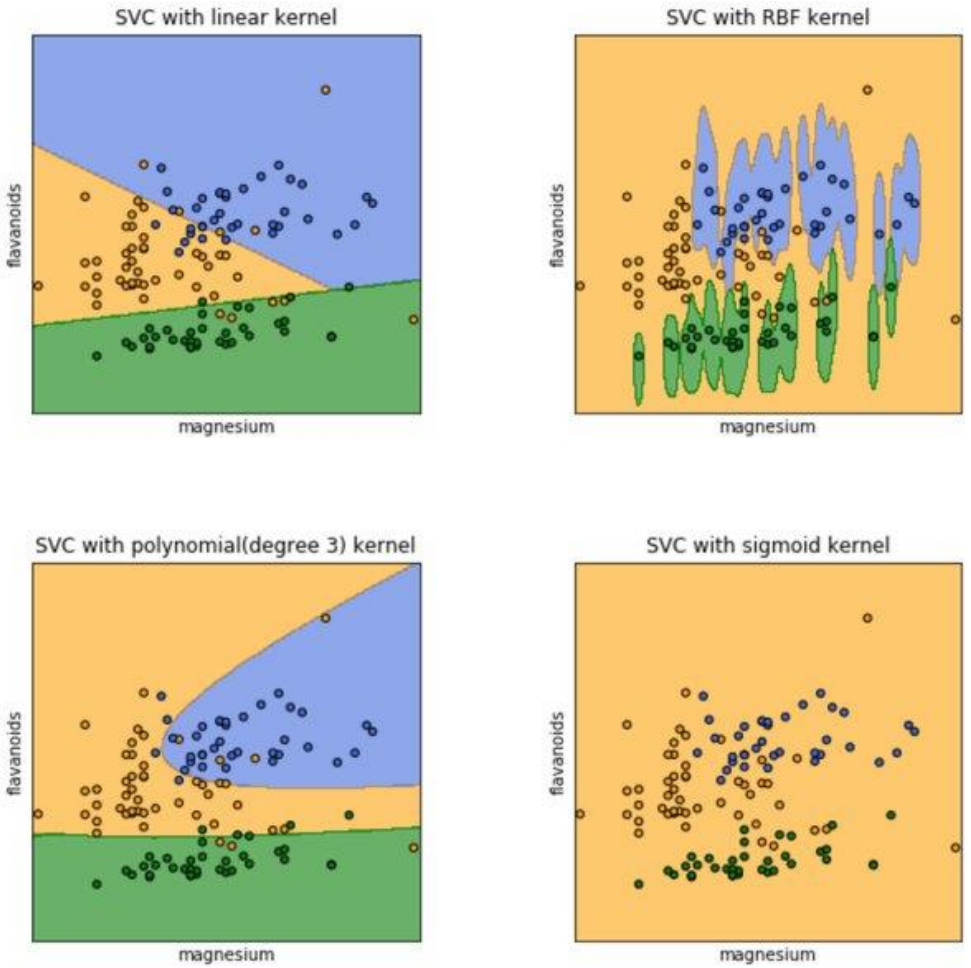
- Les calculs sont réduits
- Séparation non linéaire dans les dimensions d'origine!

(5) Noyaux de SVM

Les noyaux couramment utilisés :

- noyau linéaire,
- noyau polynomial
- noyau gaussien (RBF)
- sigmoid

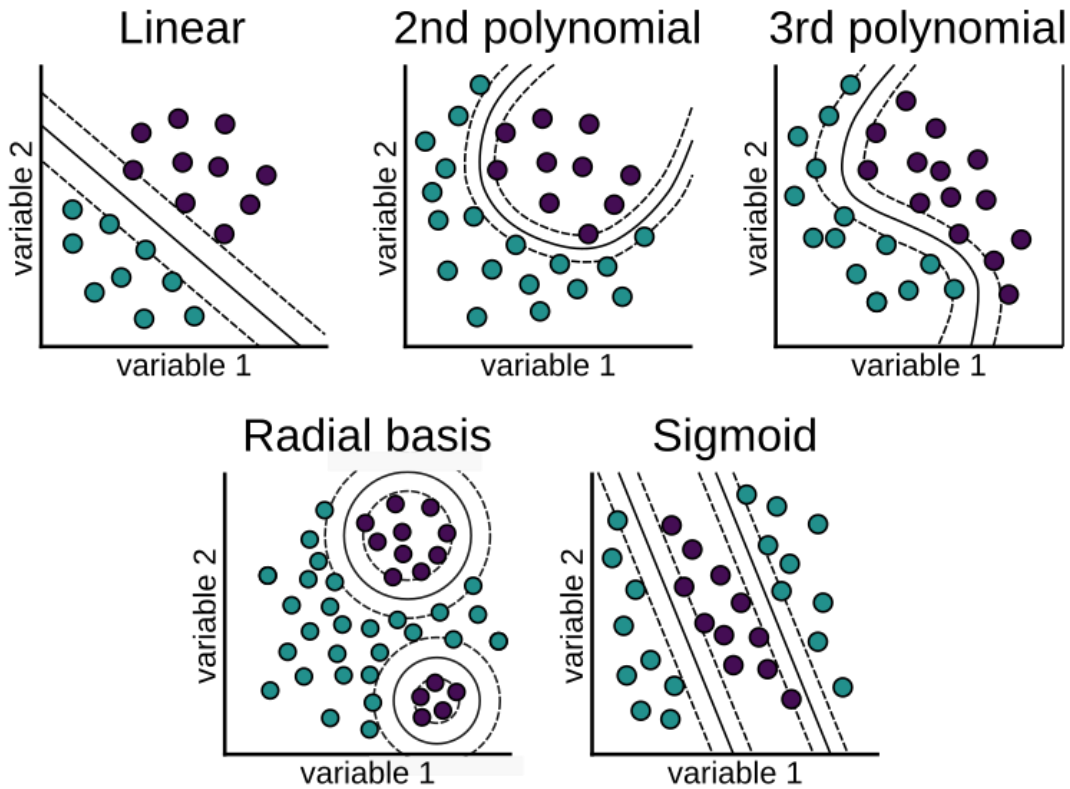




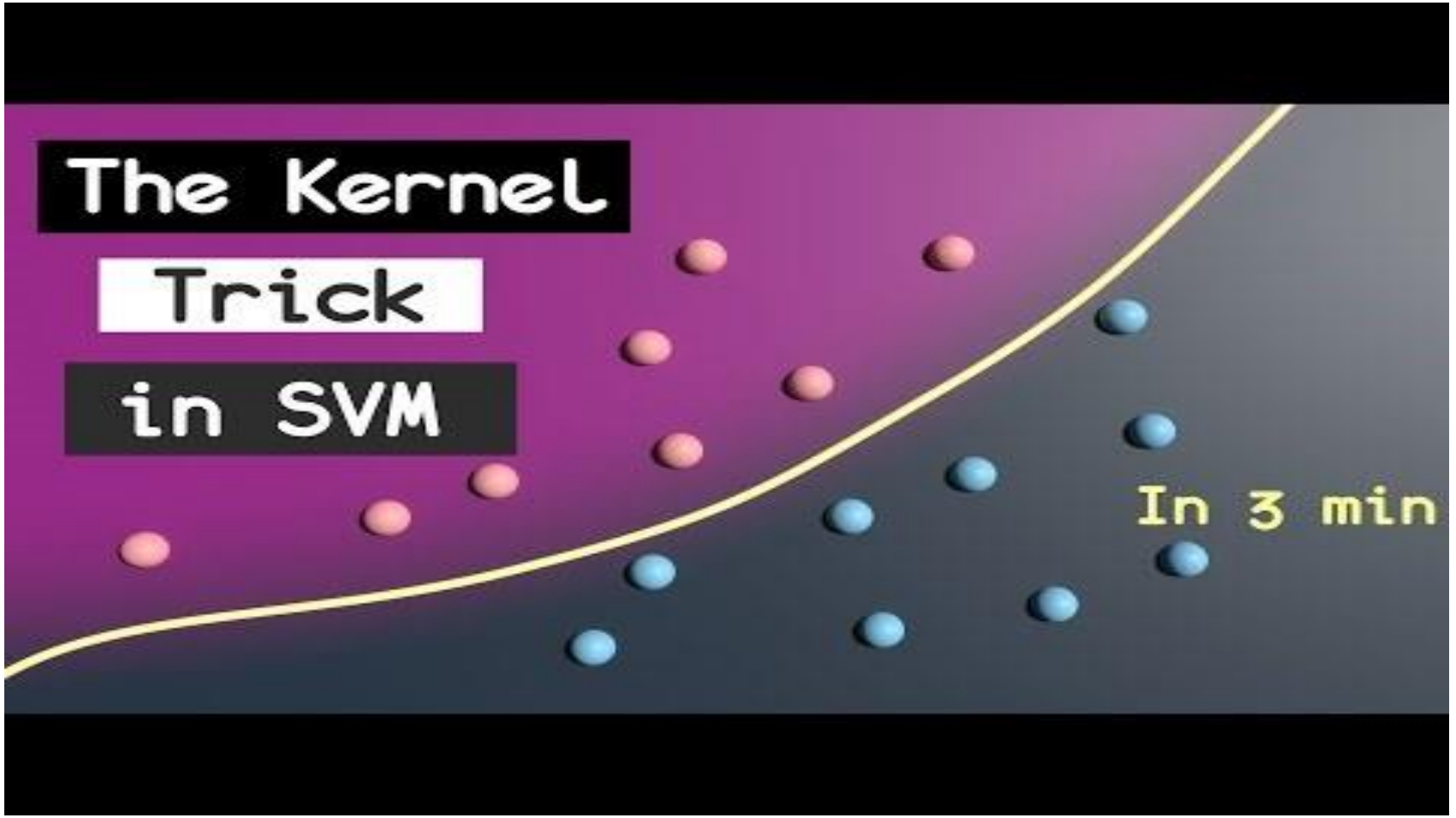
Un noyau (kernel) : fonction qui calcule le **produit scalaire** entre deux vecteurs dans l'espace de caractéristiques étendu.

(5) Noyaux de SVM

Kernel function	Expression	Parameter
Liner kernel function	$K(x_i, x_j) = x_i \cdot x_j$	
Polynomial kernel function	$K(x_i, x_j) = \left(x_i \cdot x_j + 1\right)^d$	d
Radial basis function (RBF) kernel function	$K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$	$\gamma > 0$
Sigmoid kernel function	$K\left(x_i, x_j\right) = \tanh\left(b\left(x_i, x_j\right) + c\right)$	b, c



(5) Noyaux de SVM - Astuce du Noyau (kernel Trick)



https://www.youtube.com/watch?v=Q7vT0--5VII

SVM - Résumé

https://www.youtube.com/watch?v=_YPScrckx28

Principe de Fonctionnement de SVM:

1. Construit un **hyperplan** qui sépare les deux classes
2. **Maximum de marge** entre les deux classes
3. S'appuie sur certain vecteurs (**support vectors**)
4. Paramètre **C**
5. **Frontières non linéaires grâce au noyaux (kernels)**