# EPISEN – ING3. SI

# Machine Learning

**Abdallah EL HIDALI**
Tech Lead Sita For Aircraft
abdallah.el-hidali@sita.aero

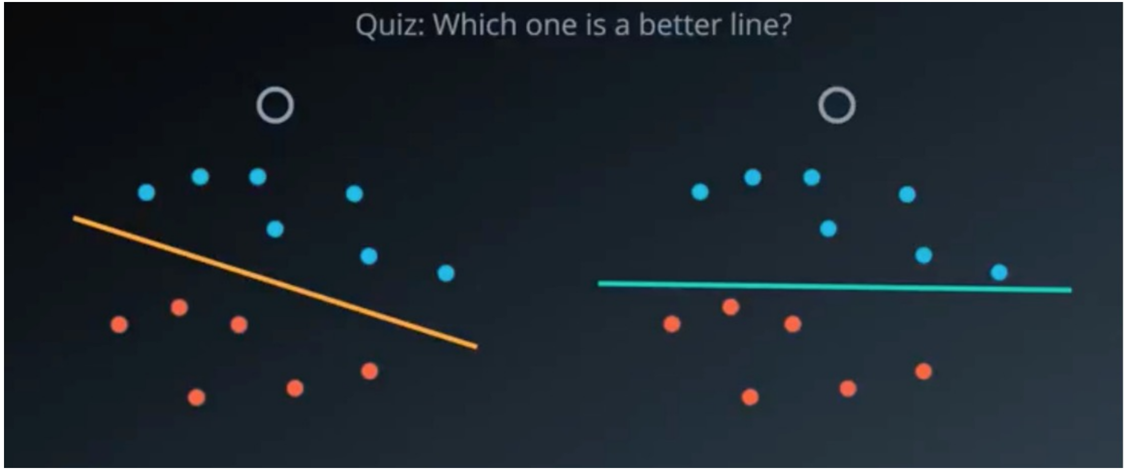
**EPISEN**
2024/2025

# VI. Support Vector Machines SVM

# Machines à vecteurs de support: Introduction

**Mise en situation:**

Considérons un plan contenant deux ensembles de points distincts, représentés respectivement en bleu et en rouge.
Deux droites sont proposées comme séparatrices de ces ensembles. Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?



# Machines à vecteurs de support: Introduction

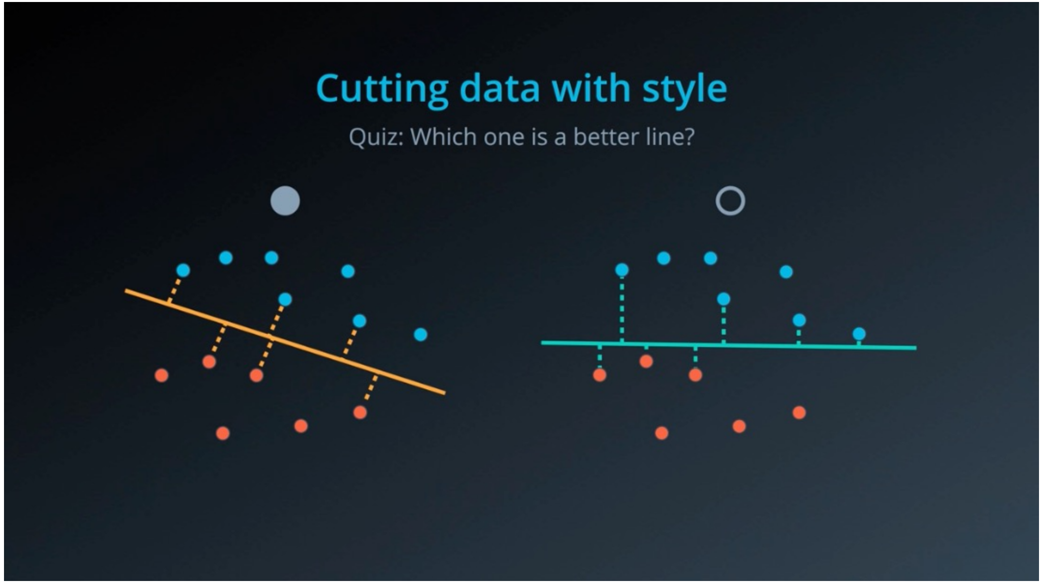
**Mise en situation:**

Considérons un plan contenant deux ensembles de points distincts, représentés respectivement en bleu et en rouge.
Deux droites sont proposées comme séparatrices de ces ensembles. Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?

# Machines à vecteurs de support: Introduction

**Mise en situation:**

Considérons un plan contenant deux ensembles de points distincts, représentés respectivement en bleu et en rouge.
Deux droites sont proposées comme séparatrices de ces ensembles. Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?



# Machines à vecteurs de support: Introduction

**Mise en situation:**

Considérons un plan contenant deux ensembles de points distincts, représentés respectivement en bleu et en rouge.
Deux droites sont proposées comme séparatrices de ces ensembles. Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?

# Machines à vecteurs de support: Introduction

**Mise en situation:**

Considérons un plan contenant deux ensembles de points distincts, représentés respectivement en bleu et en rouge.
Deux droites sont proposées comme séparatrices de ces ensembles. Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?

# Machines à vecteurs de support: Introduction

## Rappel

Dans le cadre d'un problème de classification binaire, l'objectif est de déterminer la frontière de décision optimale qui sépare les classes. Cette frontière est généralement représentée par une fonction linéaire ou non linéaire, selon la complexité du problème.
Pour trouver cette frontière optimale, on cherche à minimiser une fonction de coût, typiquement :

- **MAE (Mean Absolute Error) : L'erreur absolue moyenne**
- **MSE (Mean Squared Error) : L'erreur quadratique moyenne**

Le processus d'optimisation vise à ajuster les paramètres de la frontière de décision de manière à minimiser l'une de ces métriques d'erreur, permettant ainsi une classification plus précise des données.

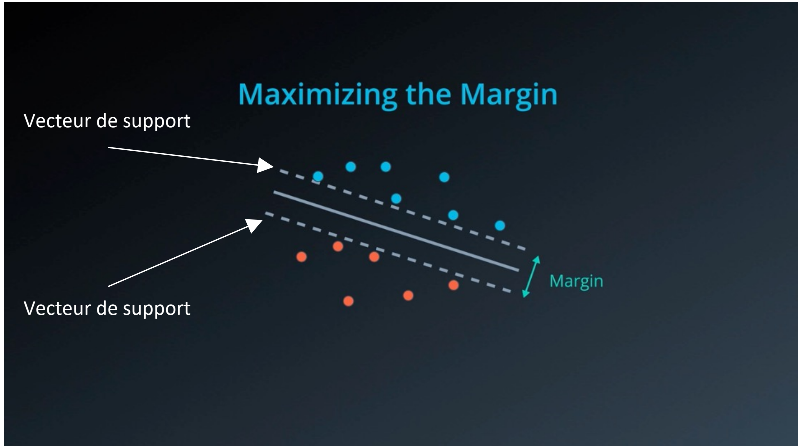
# Machines à vecteurs de support: Introduction

Dans le contexte des Machines à Vecteurs de Support (SVM), l'objectif principal est de déterminer l'hyperplan séparateur optimal qui maximise la marge entre les classes.

**Principe clé :**

L'algorithme SVM cherche à identifier l'hyperplan qui maintient la plus grande distance possible avec les points les plus proches de chaque classe, appelés vecteurs de support.

Dans le cas de données linéairement séparables en deux dimensions, cet hyperplan se manifeste comme une droite. Pour des espaces de dimensions supérieures ou des problèmes non linéaires, des techniques de noyau (kernel trick) sont employées pour projeter les données dans un espace de plus grande dimension où elles deviennent séparables.



# Machines à vecteurs de support: Introduction

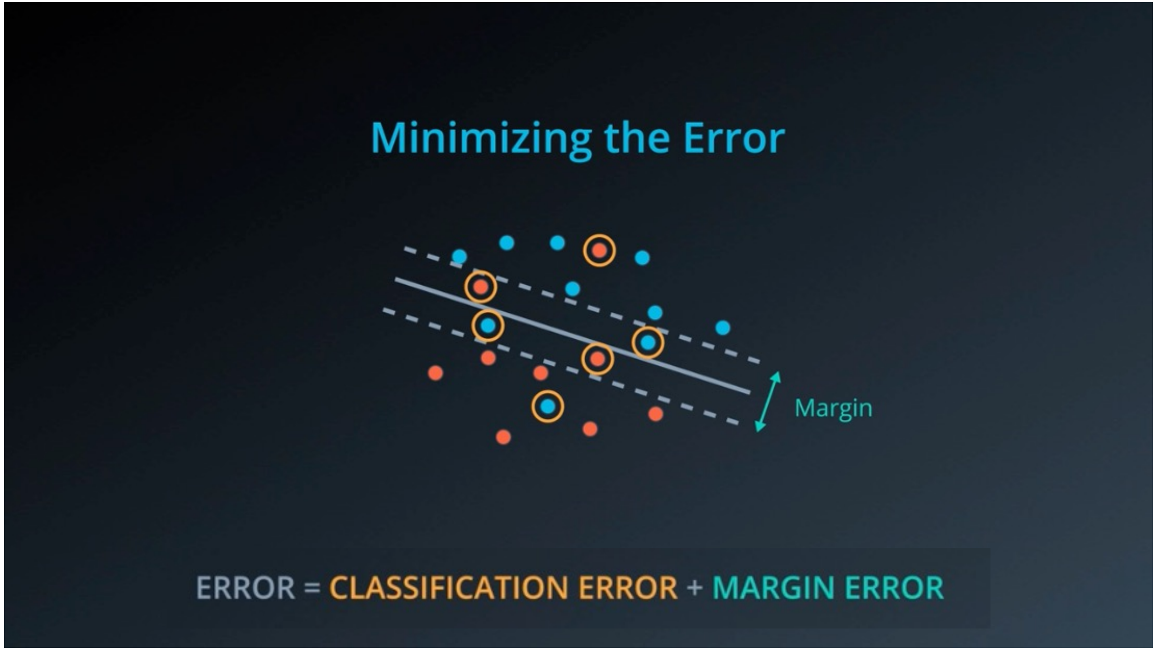
Prenons un cas non linéaire

# Machines à vecteurs de support: Introduction

L'objectif est de maximiser la marge de séparation entre les classes.

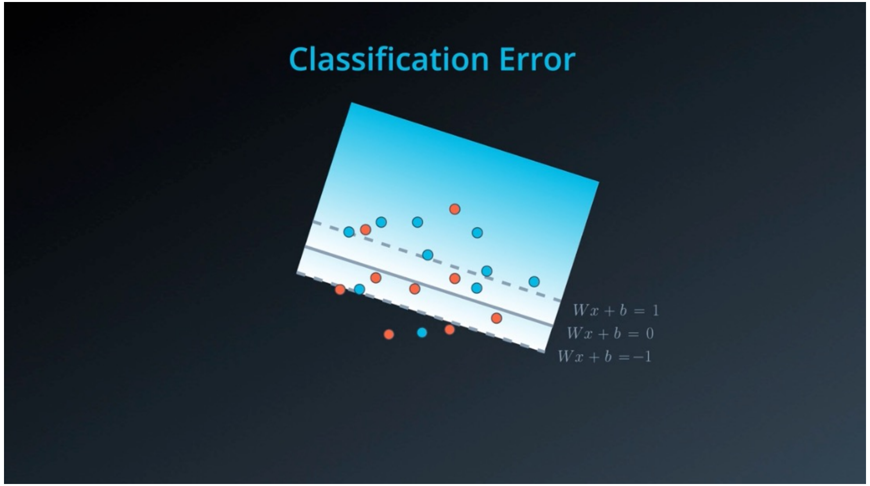
# Machines à vecteurs de support: Introduction

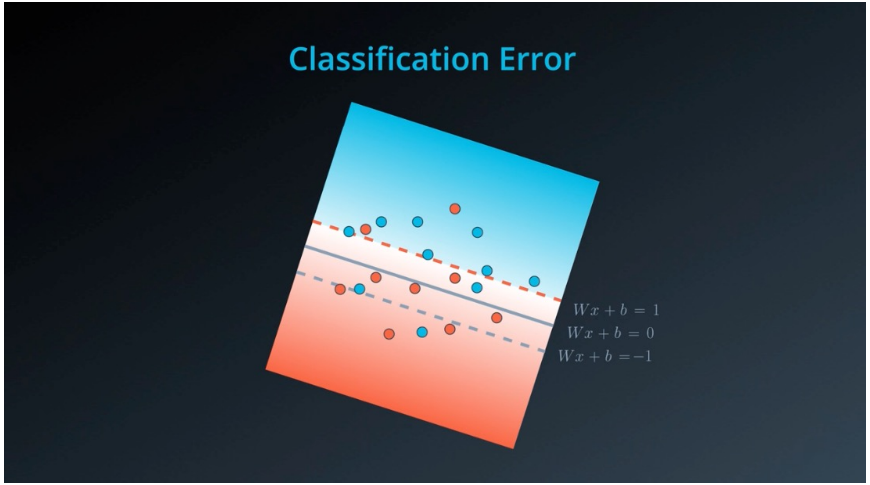
Les points mal classés et la largeur de la marge sont optimisés simultanément : on minimise l'erreur de classification tout en maximisant la marge.



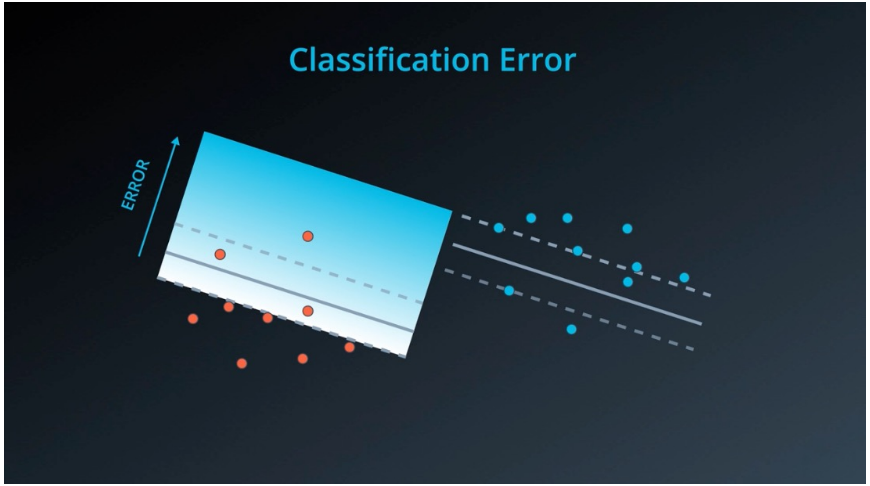
# Machines à vecteurs de support: Erreur de classification

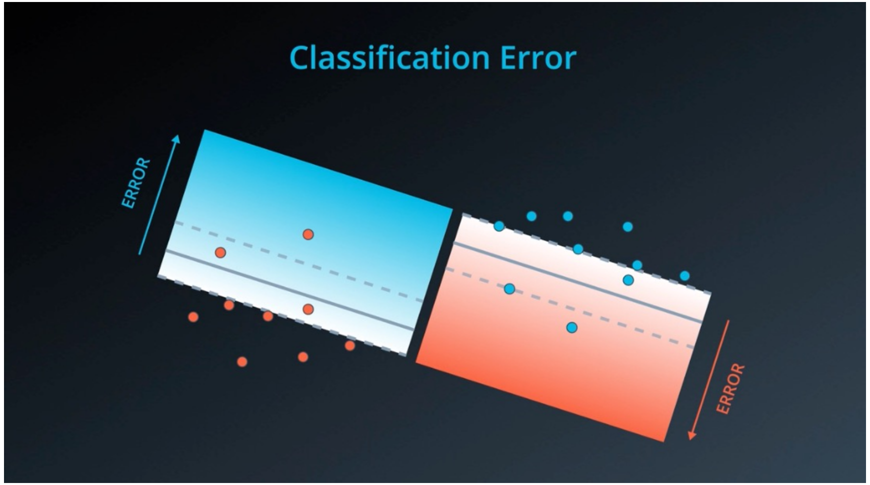
# Machines à vecteurs de support: Erreur de classification



# Machines à vecteurs de support: Erreur de classification

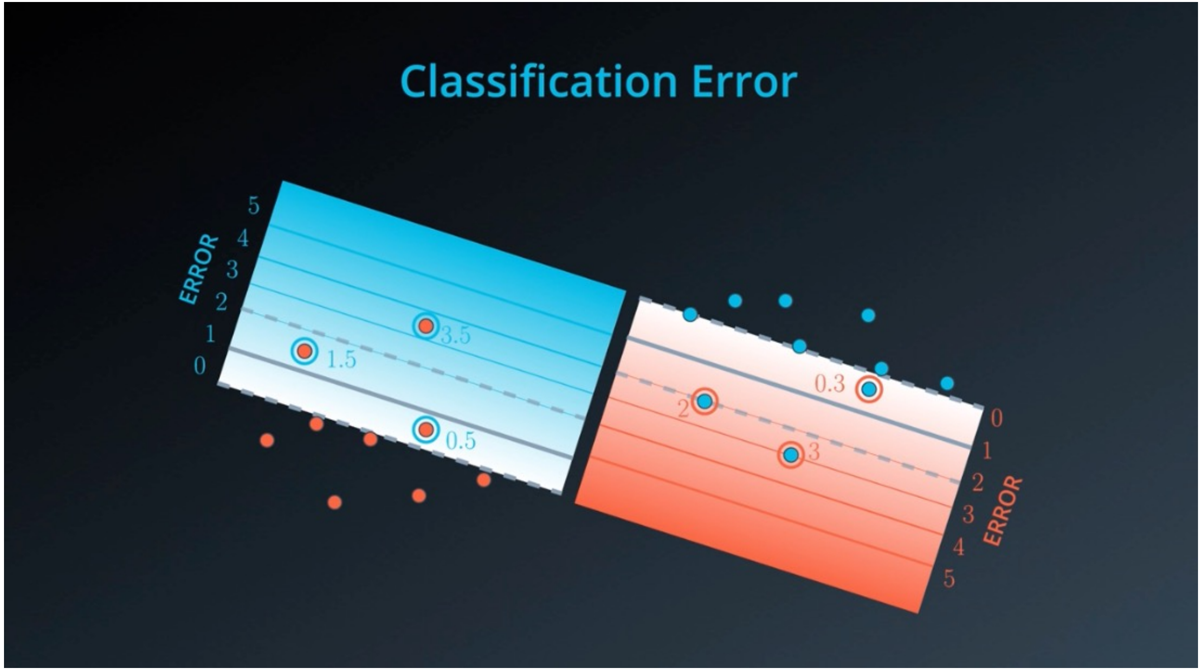


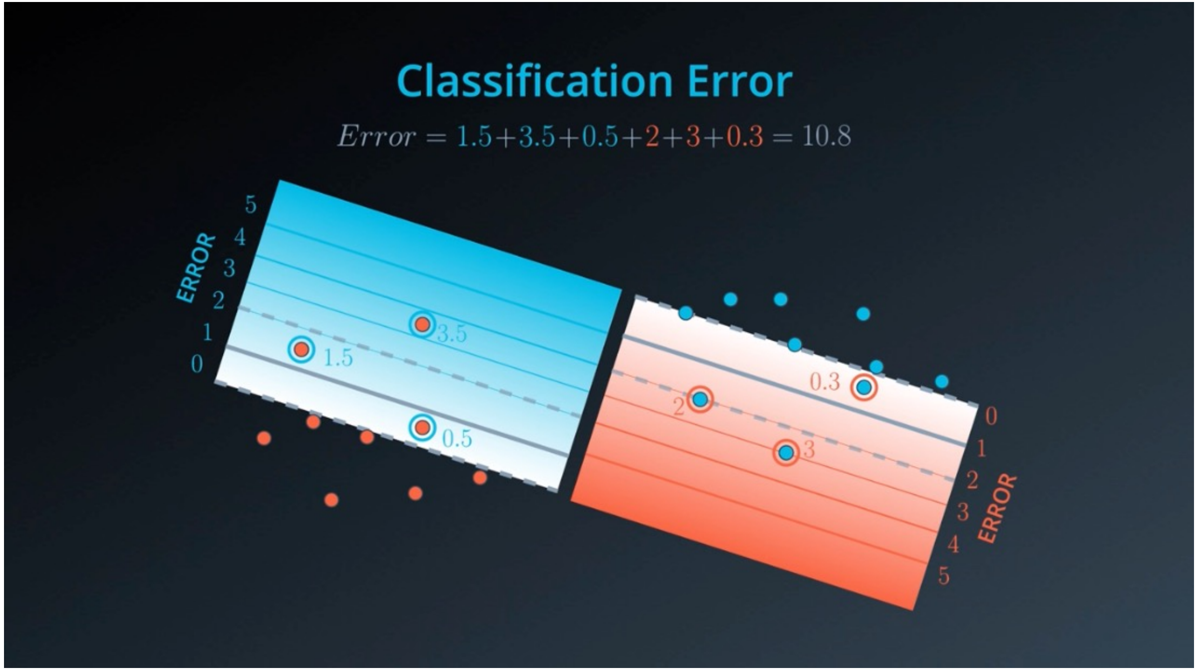
# Machines à vecteurs de support: Erreur de classification

# Machines à vecteurs de support: Erreur de classification

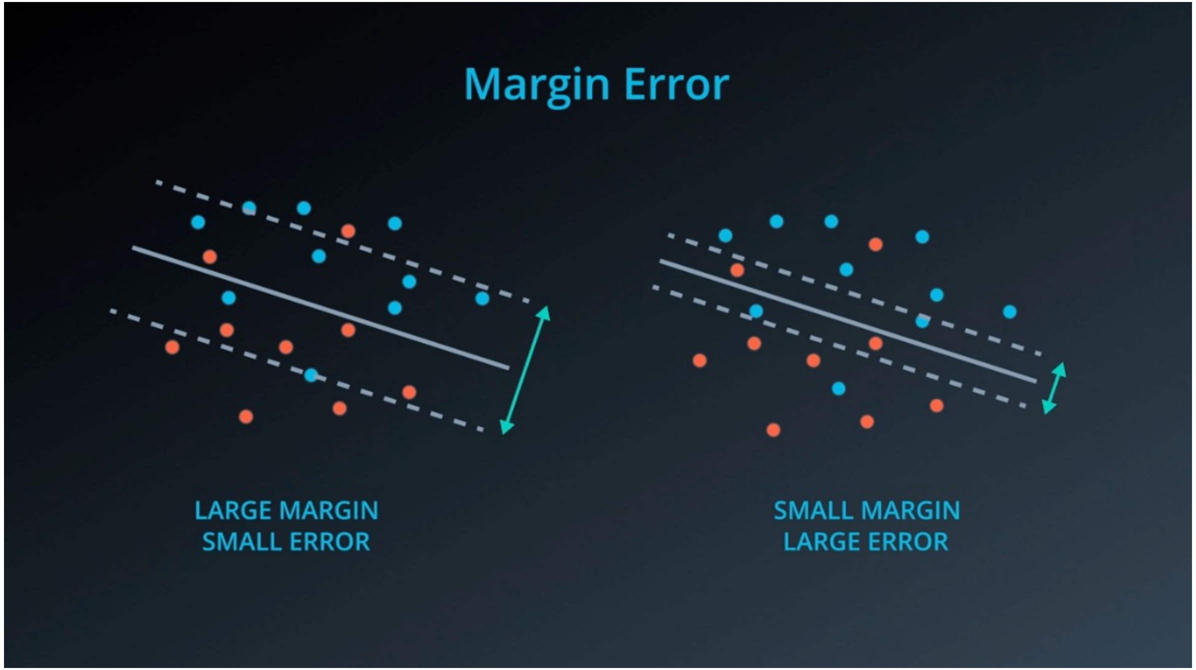
# Machines à vecteurs de support: Erreur de classification



# Machines à vecteurs de support: Erreur de classification

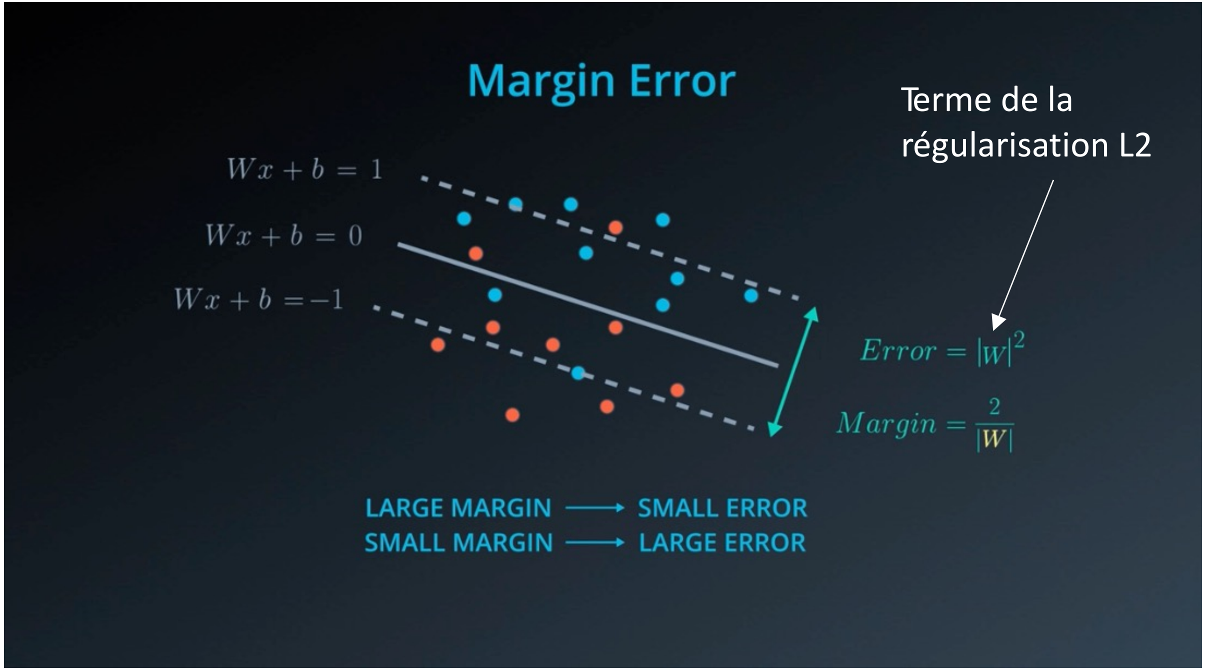


# Machines à vecteurs de support: Erreur de la marge



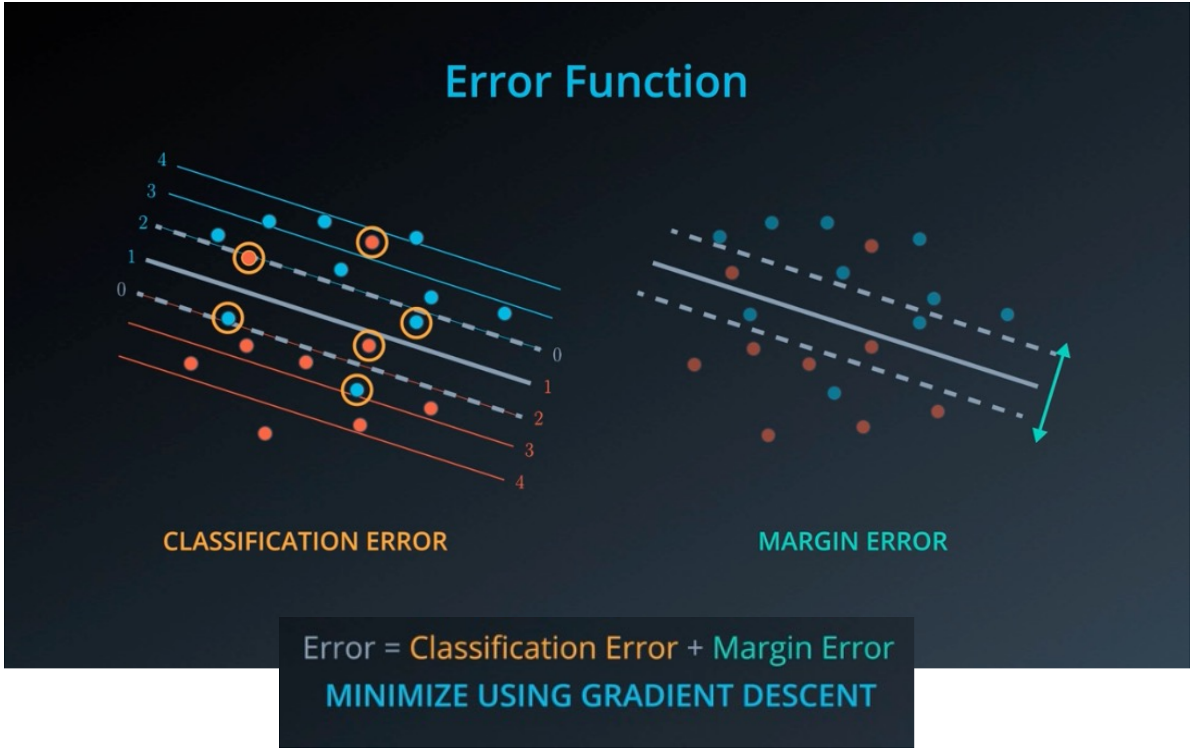

On cherche à écrire la marge comme une erreur à minimiser via la descente de gradient

# Machines à vecteurs de support: Erreur de la marge



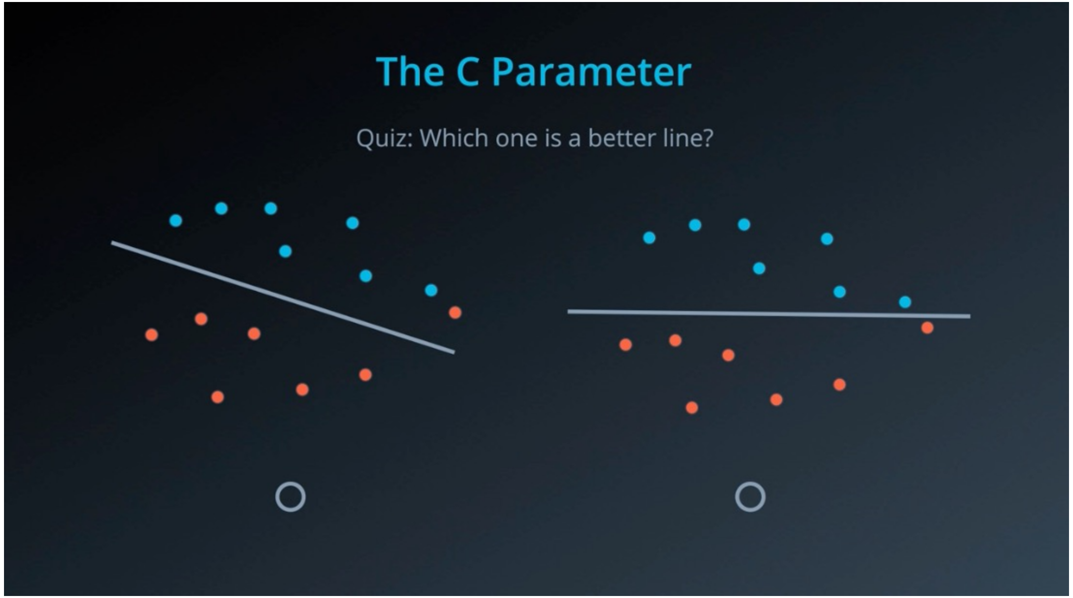

On cherche à écrire la marge comme une erreur à minimiser via la descente de gradient

# Machines à vecteurs de support: la fonction coût à minimiser



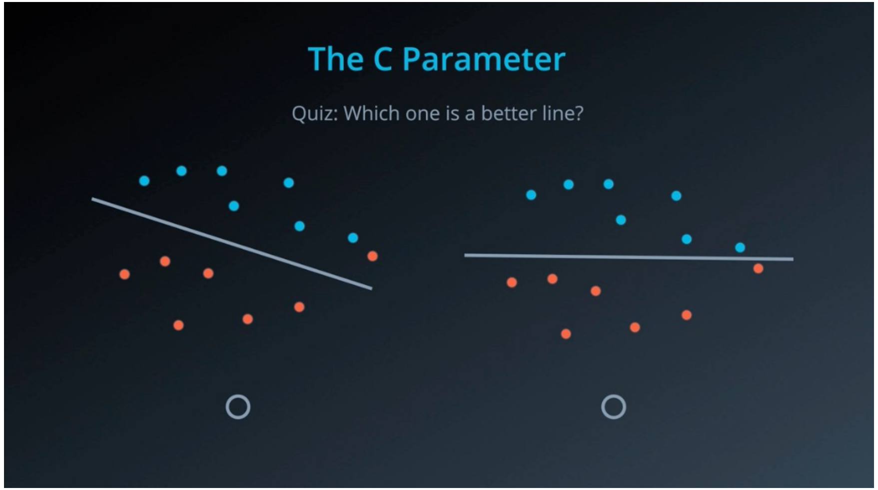

On minimise la fonction coût en utilisant la descente de gradient

# Machines à vecteurs de support: Le paramètre C



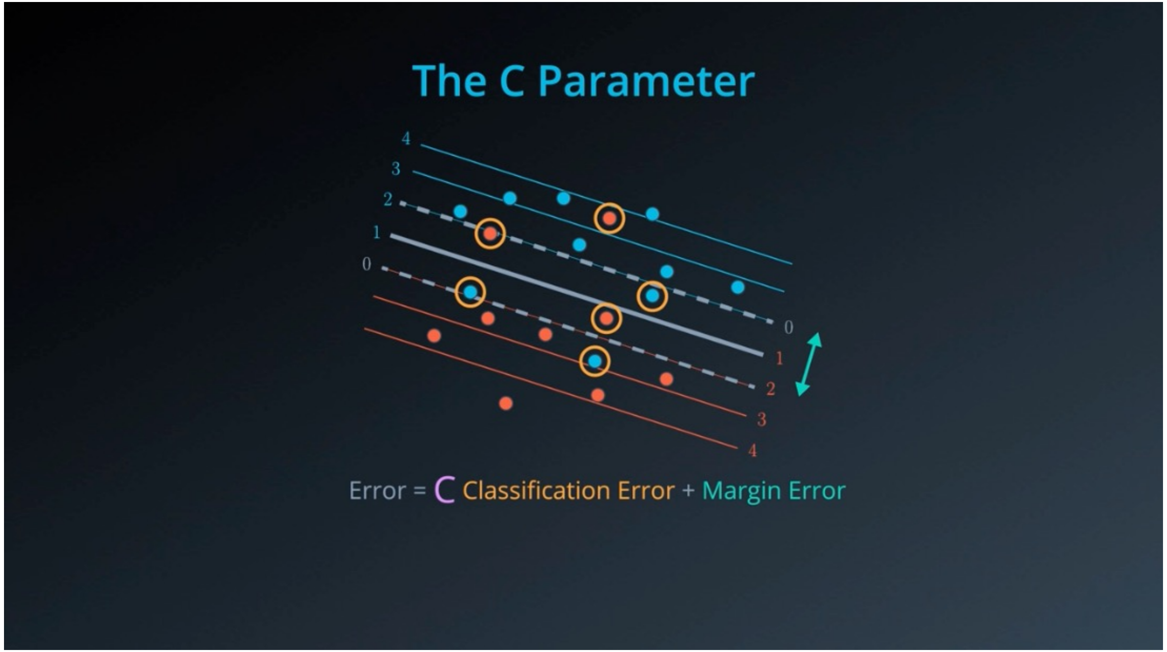

Quelle est la droite la plus appropriée pour effectuer cette séparation de manière optimale ?

# Machines à vecteurs de support: Le paramètre C



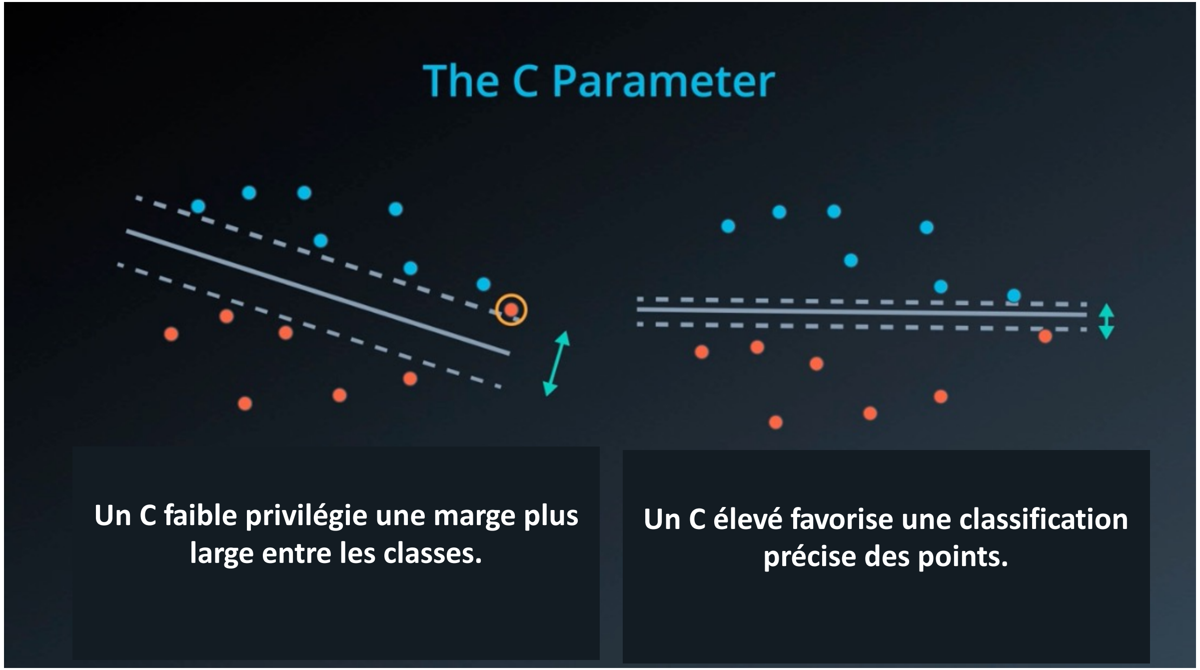

- En réalité, il n'existe pas une seule bonne réponse, car le choix du modèle dépend de l'objectif spécifique de l'analyse.
- On peut préférer un modèle sans erreur de classification ou opter pour celui qui maximise la marge entre les classes malgré quelques erreurs.
- Cette flexibilité est ajustable grâce au paramètre C qui permet d'équilibrer ces deux aspects.

# Machines à vecteurs de support: Le paramètre C

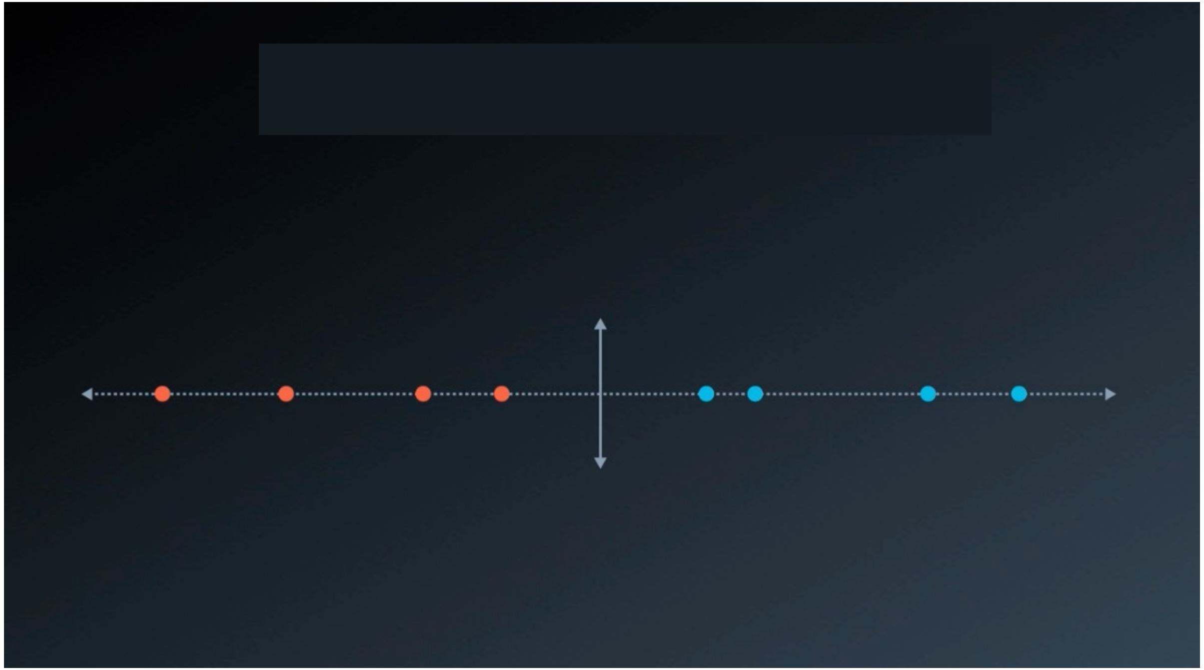



- ✓ **Un C élevé favorise une classification précise des points.**
- ✓ **Un C faible privilégie une marge plus large entre les classes.**

# Machines à vecteurs de support: Le paramètre C

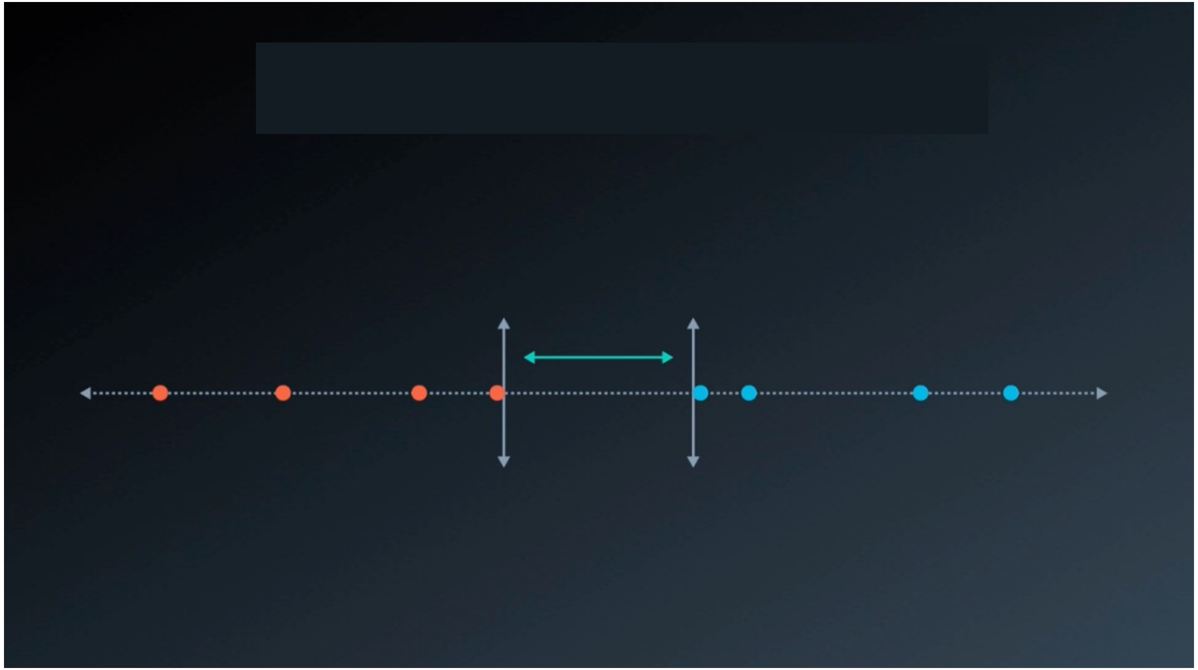


# Machines à vecteurs de support: Le noyau polynomial



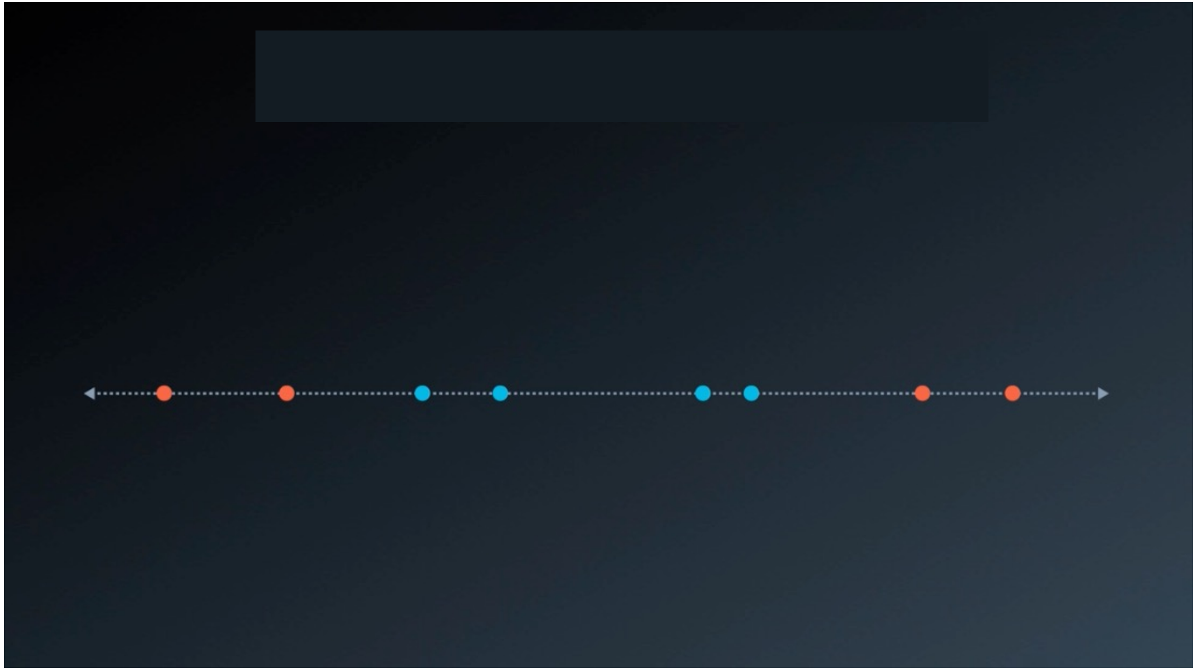
**Une simple ligne verticale suffit pour séparer les deux classes dans ce problème de classification.**

# Machines à vecteurs de support: Le noyau polynomial



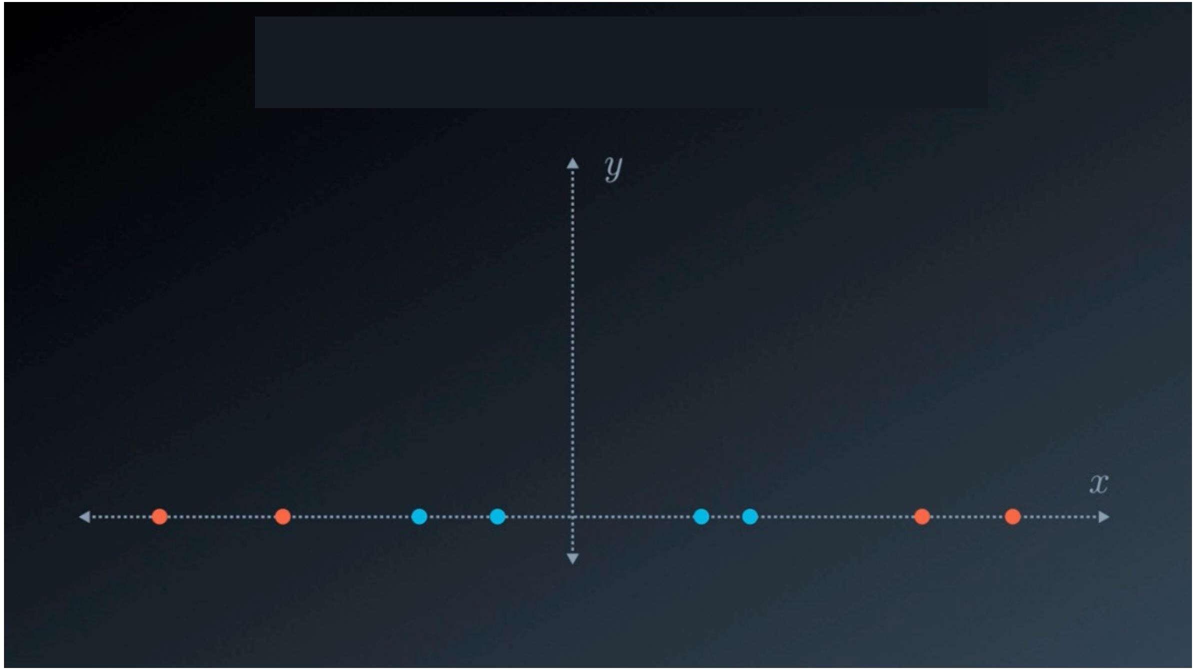
**Avec les Machines à vecteurs de support, nous allons chercher la ligne verticale qui maximise la marge entre les deux classes.**

# Machines à vecteurs de support: Le noyau polynomial



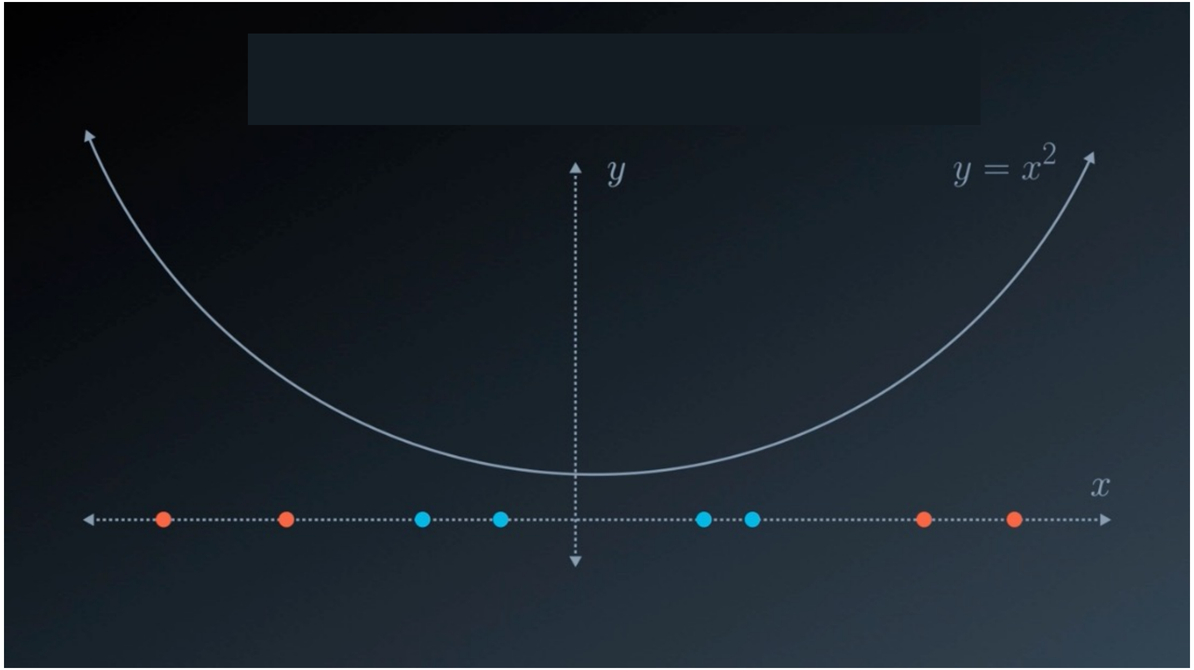
**Ce problème de classification est non linéaire et ne peut être résolu par une simple ligne droite séparant les deux classes.**

# Machines à vecteurs de support: Le noyau polynomial



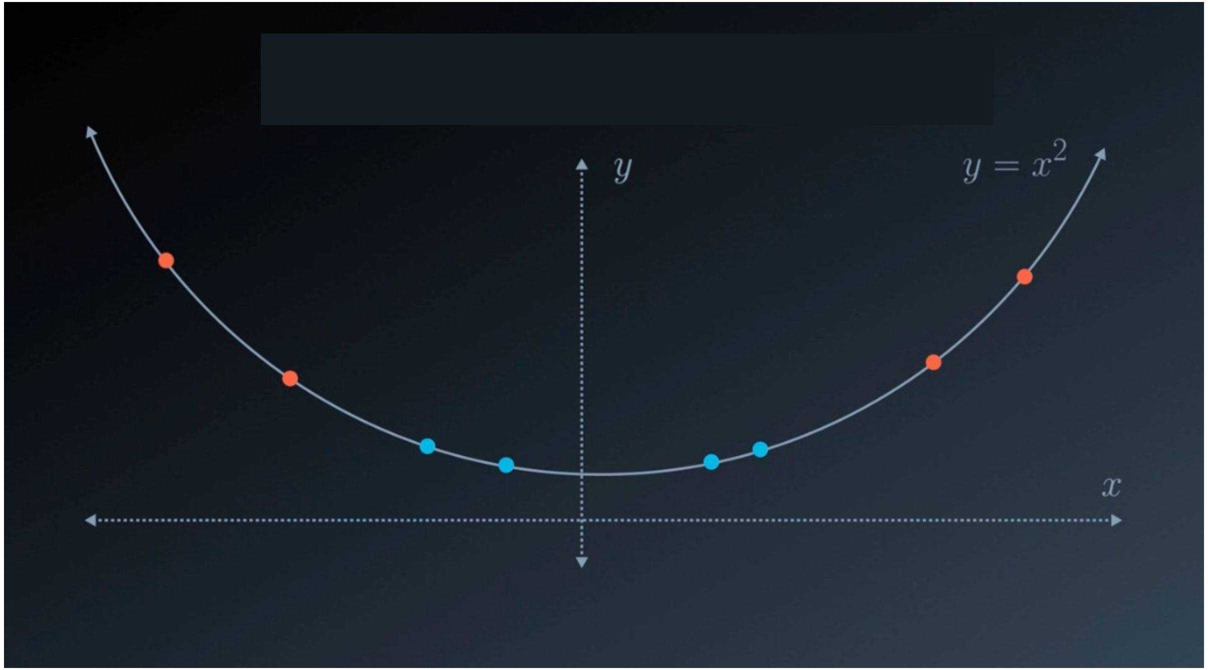

**Pour résoudre ce problème de classification, nous passons d'un plan 1D à un plan 2D en introduisant l'axe Y.**

# Machines à vecteurs de support: Le noyau polynomial



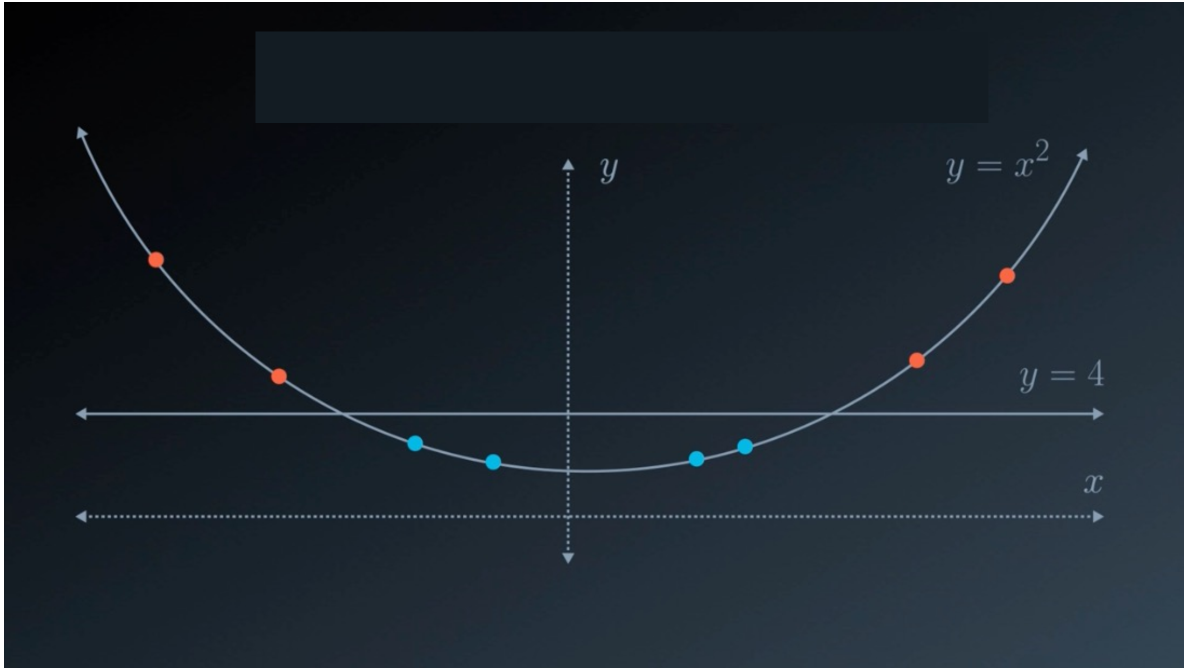

**Ensuite, on trace une parabole d'équation $y = x^2$**

# Machines à vecteurs de support: Le noyau polynomial



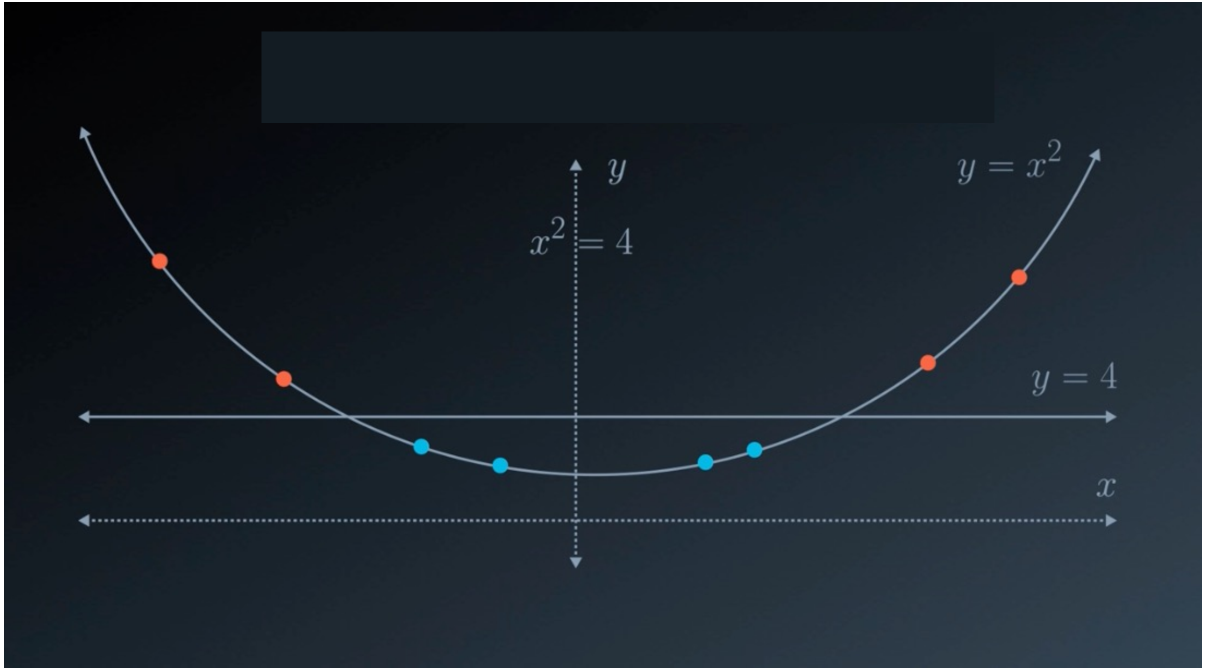

**Ensuite, on projette les points à classer sur la parabole tracée.**

# Machines à vecteurs de support: Le noyau polynomial



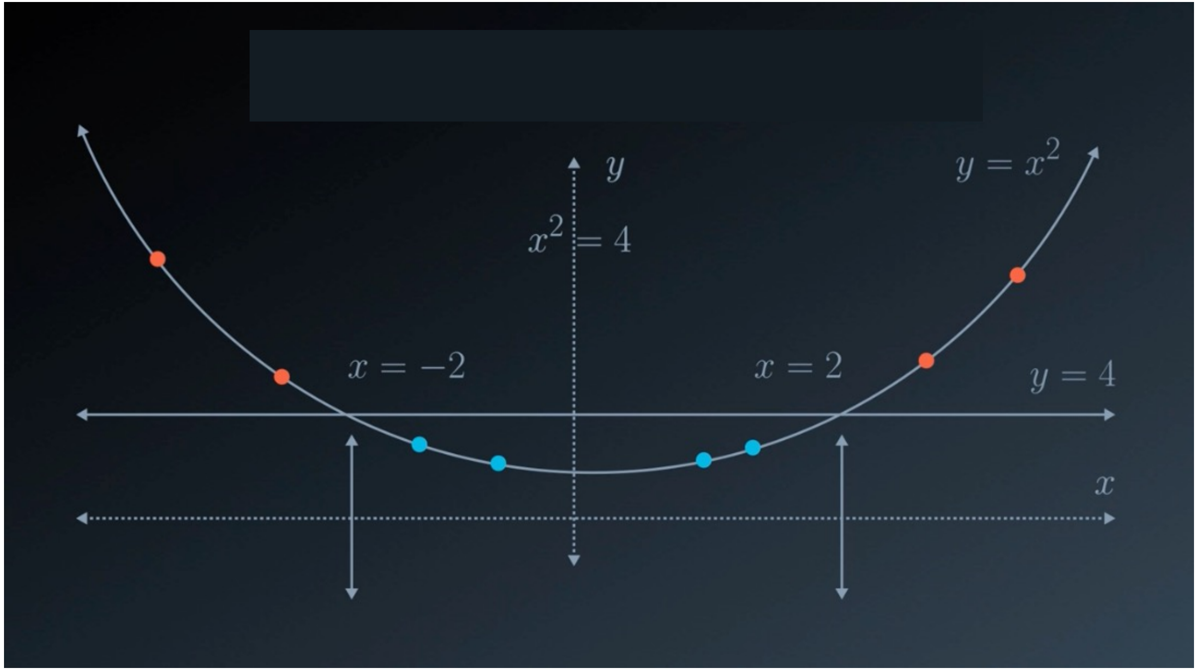

**<u>La droite</u> y = 4 permet désormais de séparer les points bleus et rouges.**

# Machines à vecteurs de support: Le noyau polynomial




**La question est maintenant de savoir comment utiliser ce résultat pour revenir à l'état initial du problème.**
**Cela commence par la résolution de l'équation $x^2 = 4$.**

# Machines à vecteurs de support: Le noyau polynomial

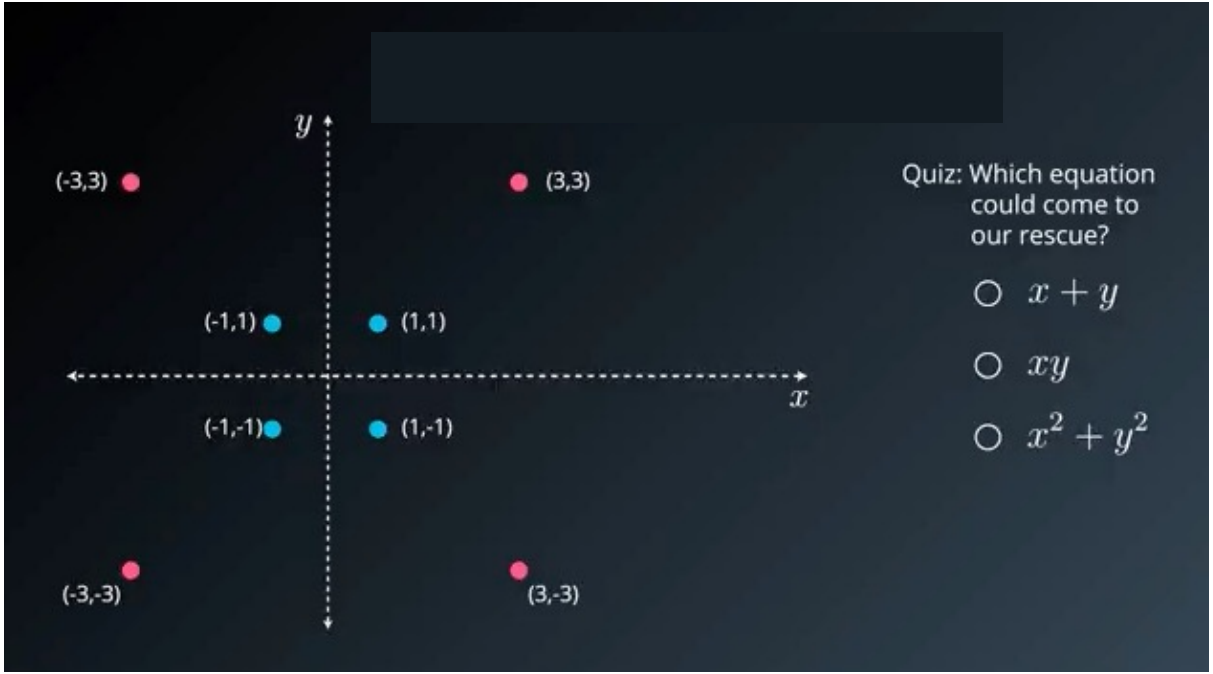



**La solution de l'équation $x^2 = 4$ est x=-2 ou x=2.**

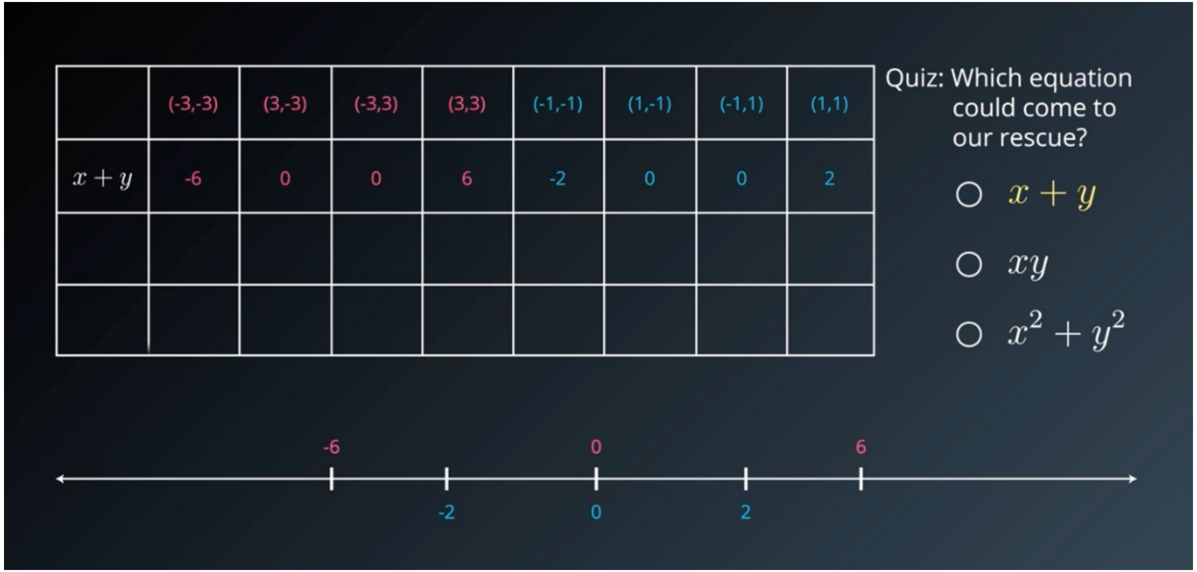
# Machines à vecteurs de support: Le noyau polynomial


- La solution x = -2 ou x = 2 de l'équation $x^2 = 4$ définit la frontière de décision pour notre problème de classification.
- Ce changement de dimension a permis d'établir cette frontière.
- Cette approche est couramment utilisée dans les machines à vecteurs de support et d'autres algorithmes d'apprentissage automatique.

# Machines à vecteurs de support: Le noyau polynomial

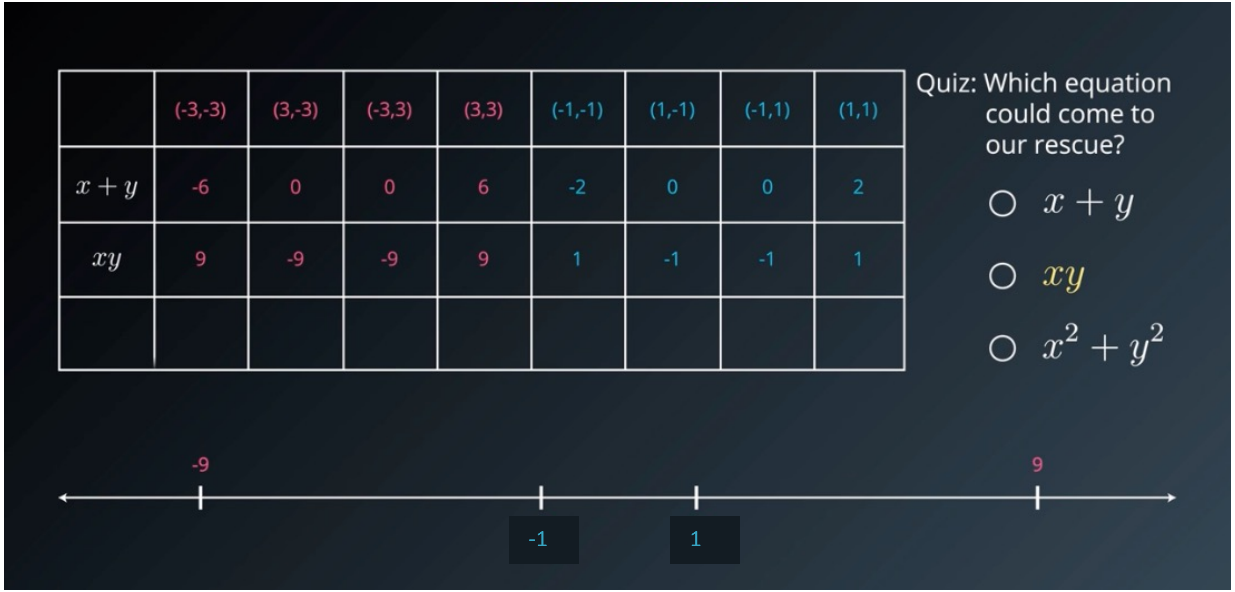


# Machines à vecteurs de support: Le noyau polynomial




- **L'équation x + y ne permet pas de séparer les points bleus des points rouges dans ce cas.**
- **Ce résultat était prévisible, car les deux classes ne sont pas linéairement séparables.**

# Machines à vecteurs de support: Le noyau polynomial



| | (-3,-3) | (3,-3) | (-3,3) | (3,3) | (-1,-1) | (1,-1) | (-1,1) | (1,1) |
|---|---|---|---|---|---|---|---|---|
| $x + y$ | -6 | 0 | 0 | 6 | -2 | 0 | 0 | 2 |
| $xy$ | 9 | -9 | -9 | 9 | 1 | -1 | -1 | 1 |
| | | | | | | | | |



**Même si xy est une opération non linéaire, cette équation ne permet pas de séparer les points bleus des points rouges dans ce cas.**

# Machines à vecteurs de support: Le noyau polynomial

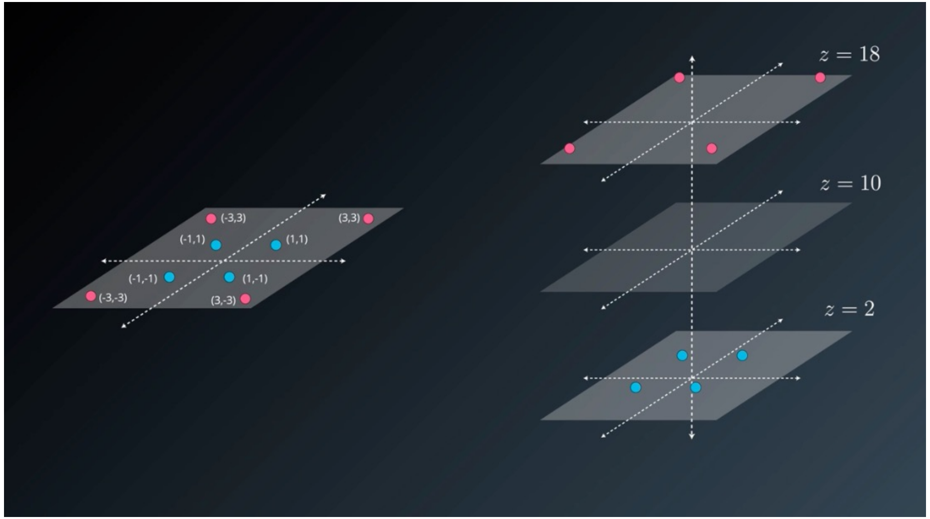
| | (-3,-3) | (3,-3) | (-3,3) | (3,3) | (-1,-1) | (1,-1) | (-1,1) | (1,1) |
|---|---|---|---|---|---|---|---|---|
| $x+y$ | -6 | 0 | 0 | 6 | -2 | 0 | 0 | 2 |
| $xy$ | 9 | -9 | -9 | 9 | 1 | -1 | -1 | 1 |
| $x^2+y^2$ | 18 | 18 | 18 | 18 | 2 | 2 | 2 | 2 |



- **L'opération non linéaire $x^2 + y^2$ permet de séparer les points bleus des points rouges dans ce cas.**
- **Les points rouges ont pour équation $x^2 + y^2 = 18$**
- **Les points bleus ont pour équation $x^2 + y^2 = 2$**
- **Le cercle d'équation $x^2 + y^2 = 10$ peut servir de frontière pour séparer les points bleus des points rouges.**

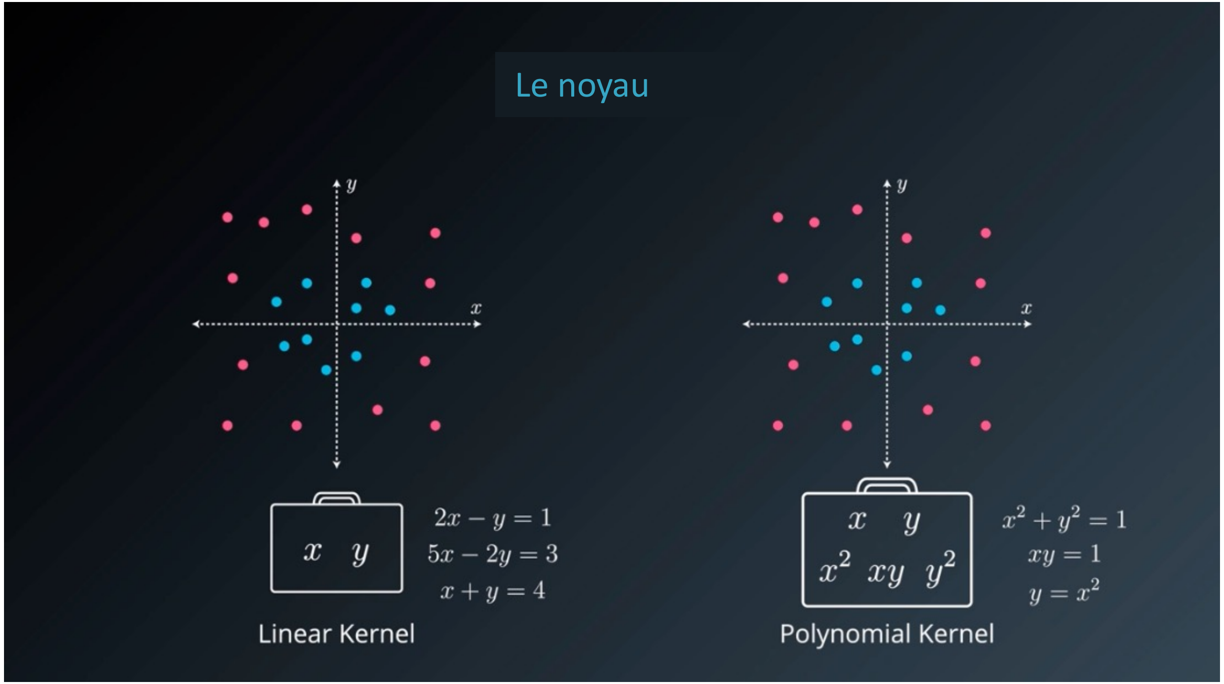
# Machines à vecteurs de support: Le noyau polynomial

# Machines à vecteurs de support: Le noyau polynomial



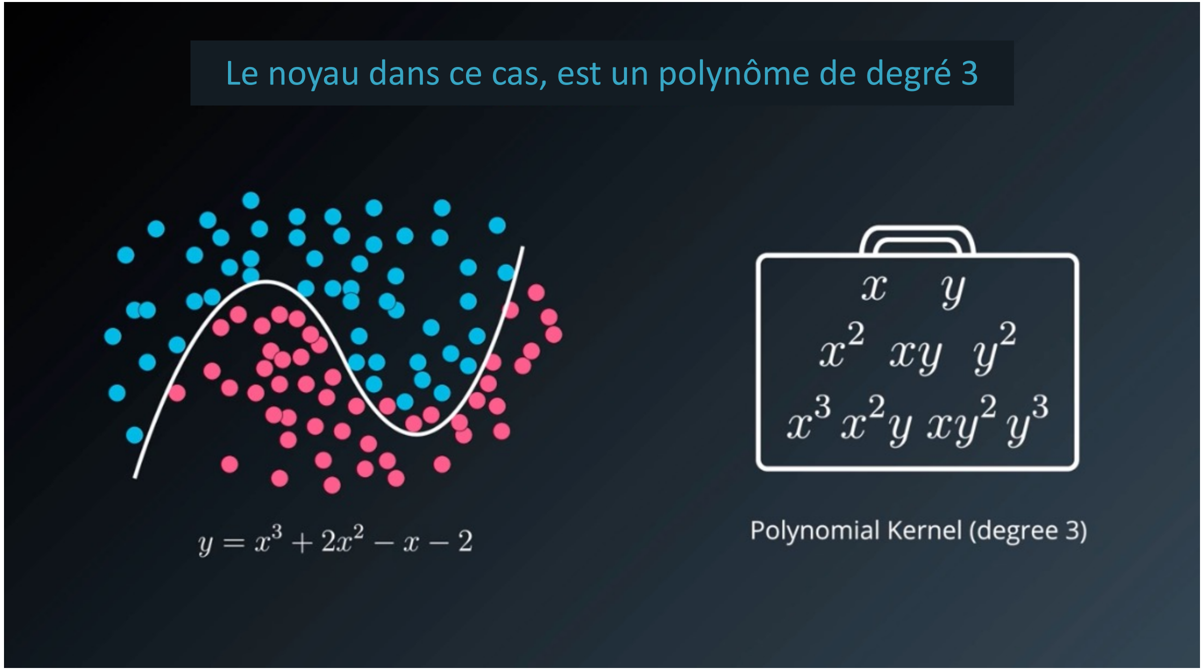

- L'algorithme SVM projette les données dans un espace de dimension supérieure où elles deviennent linéairement séparables
- Cette transformation est réalisée implicitement grâce à des fonctions noyau
- Une fois les données séparées dans cet espace de haute dimension, l'algorithme détermine un hyperplan optimal qui maximise la marge entre les classes
- La décision de classification est ensuite ramenée à l'espace d'origine, ce qui se traduit par une frontière de décision non linéaire dans l'espace initial

# Machines à vecteurs de support: Le noyau polynomial



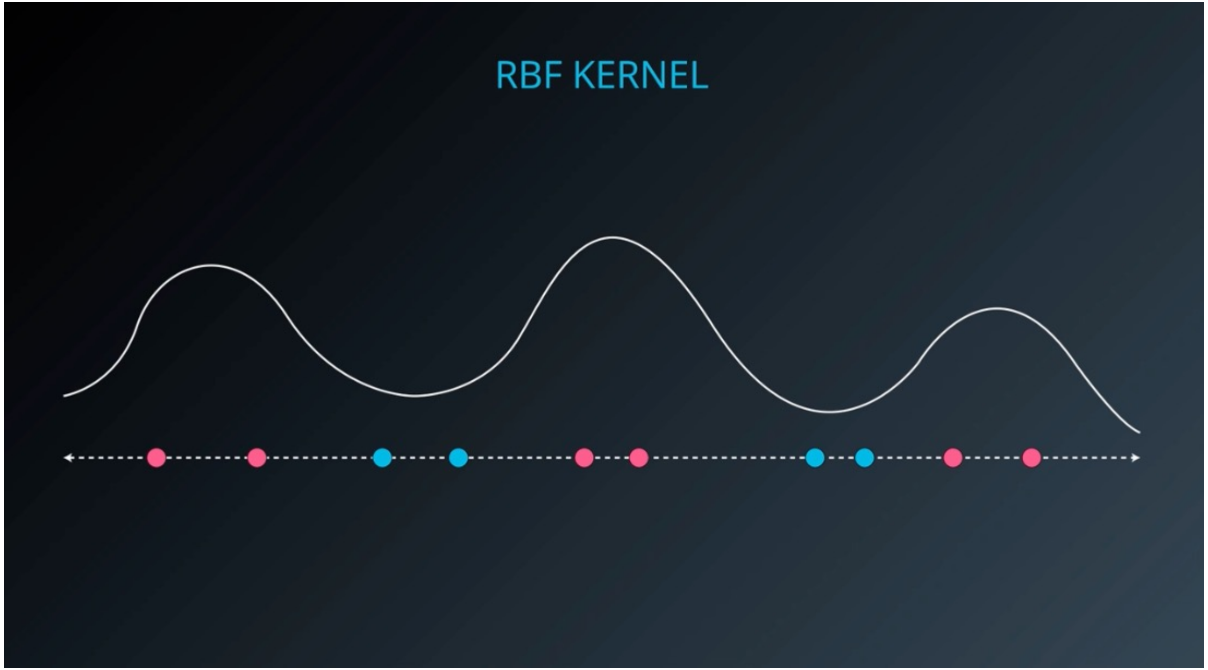

- **Le noyau (kernel) dans les machines à vecteurs de support, est une fonction qui transforme implicitement les données en un espace de dimension supérieure, permettant de résoudre des problèmes de classification non linéaires.**

# Machines à vecteurs de support: Le noyau polynomial



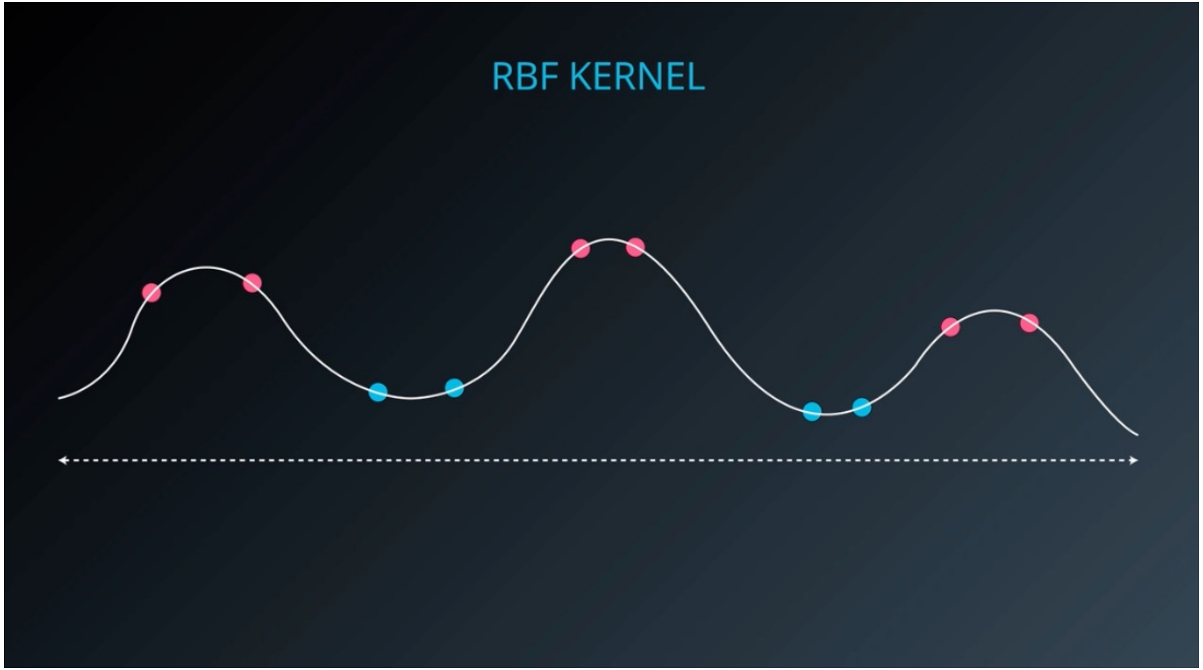
# Machines à vecteurs de support: Le noyau gaussien (RBF)

*RBF (Radial Basis Function)*



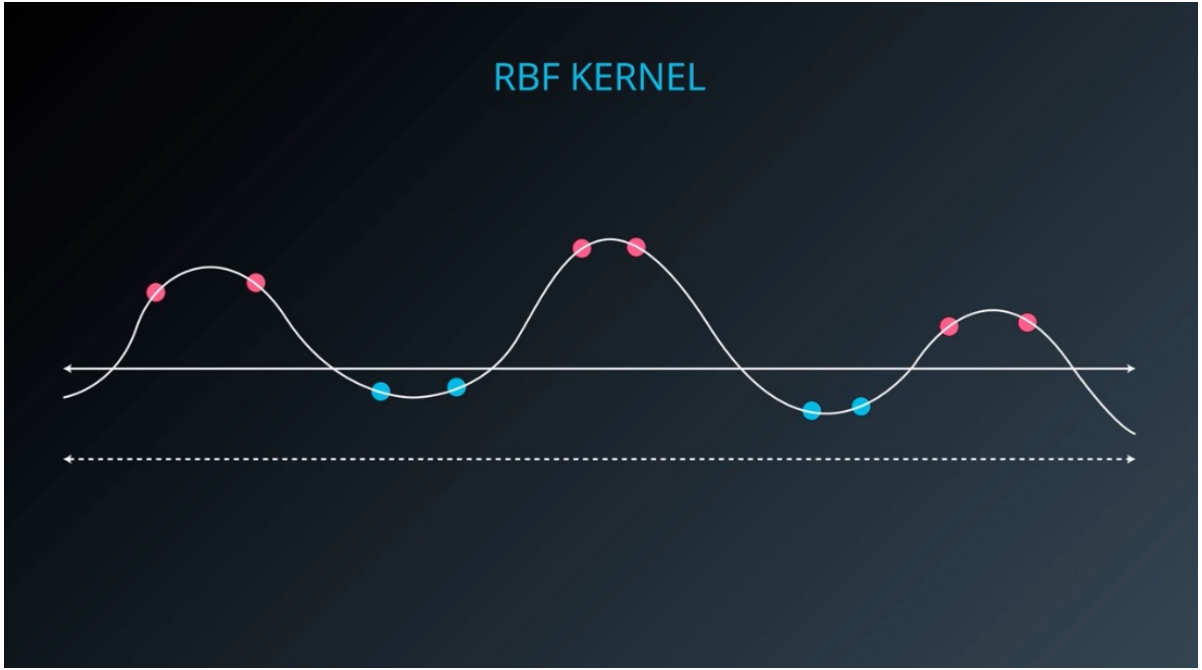

**Dans le cadre d'un noyau RBF, l'objectif est de séparer les points bleus des points rouges en utilisant une série de distributions gaussiennes. On positionne les gaussiennes de manière à ce que leurs sommets coïncident avec les points rouges, tandis que les points bleus se retrouvent dans les creux entre ces gaussiennes.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



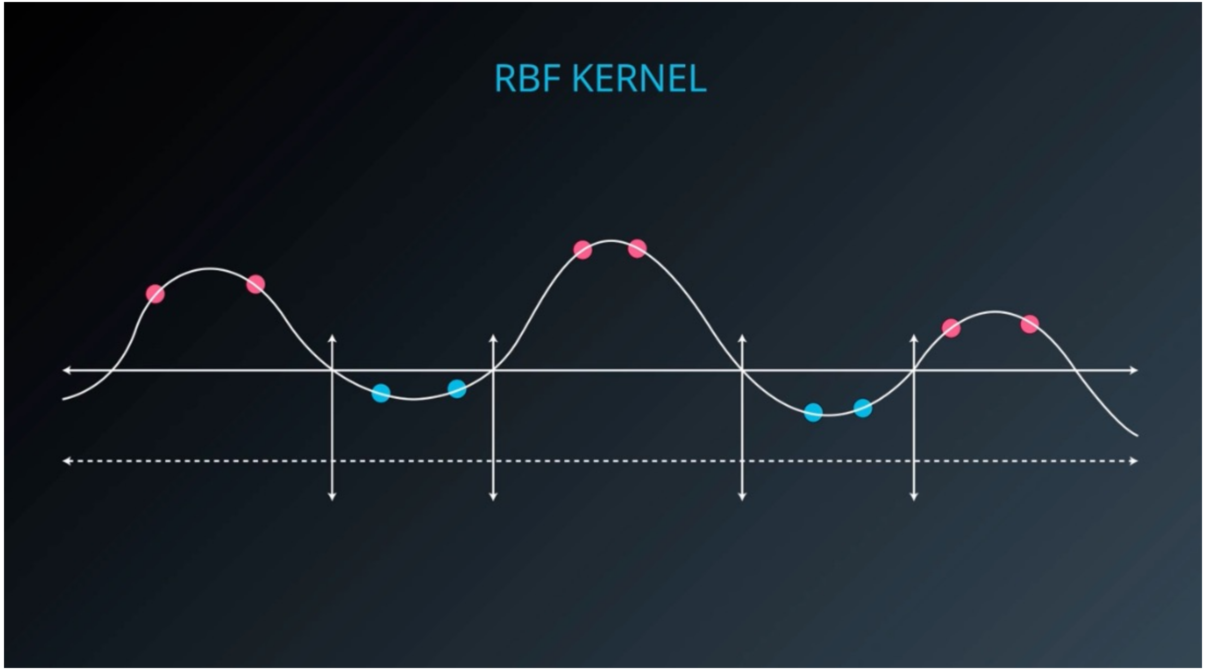

**On projette ensuite les points bleus et rouges sur cette série de gaussiennes.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



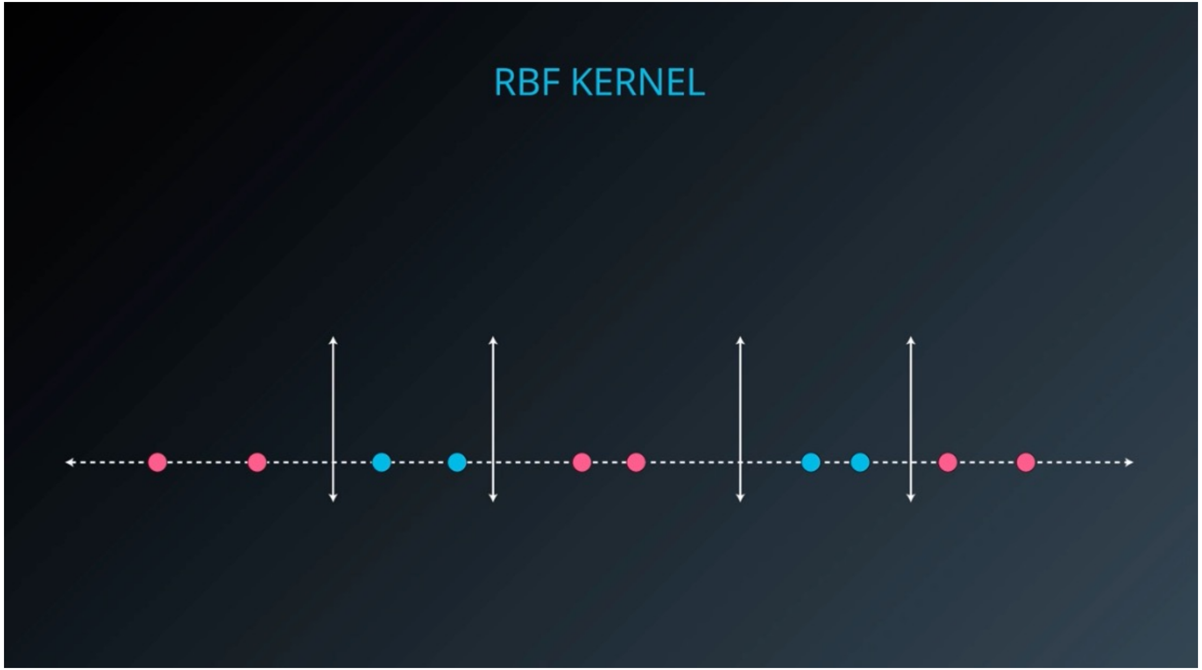

**Ensuite, on sépare les points bleus et rouges projetés par une droite.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)




**On identifie alors les points d'intersection entre la droite de séparation et la série de gaussiennes.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)

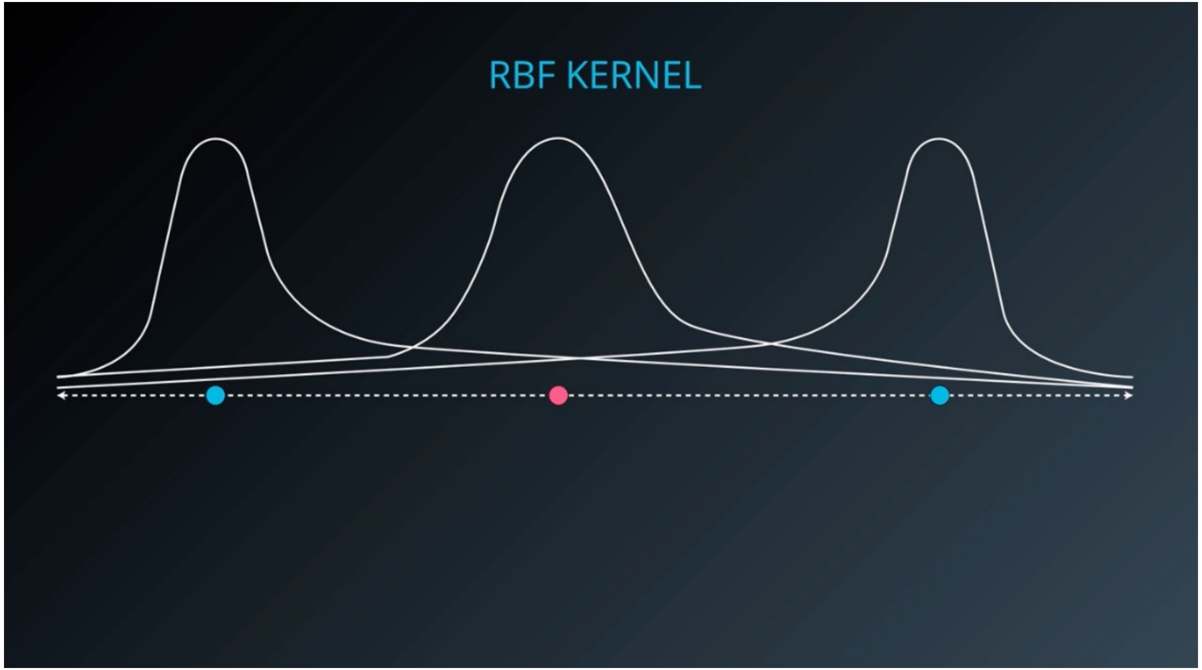



**En revenant à l'espace d'origine, les points d'intersection identifiés forment la frontière de décision.**

**Comment trouver la série de gaussiennes qui a permis d'identifier notre frontière de décision ?**

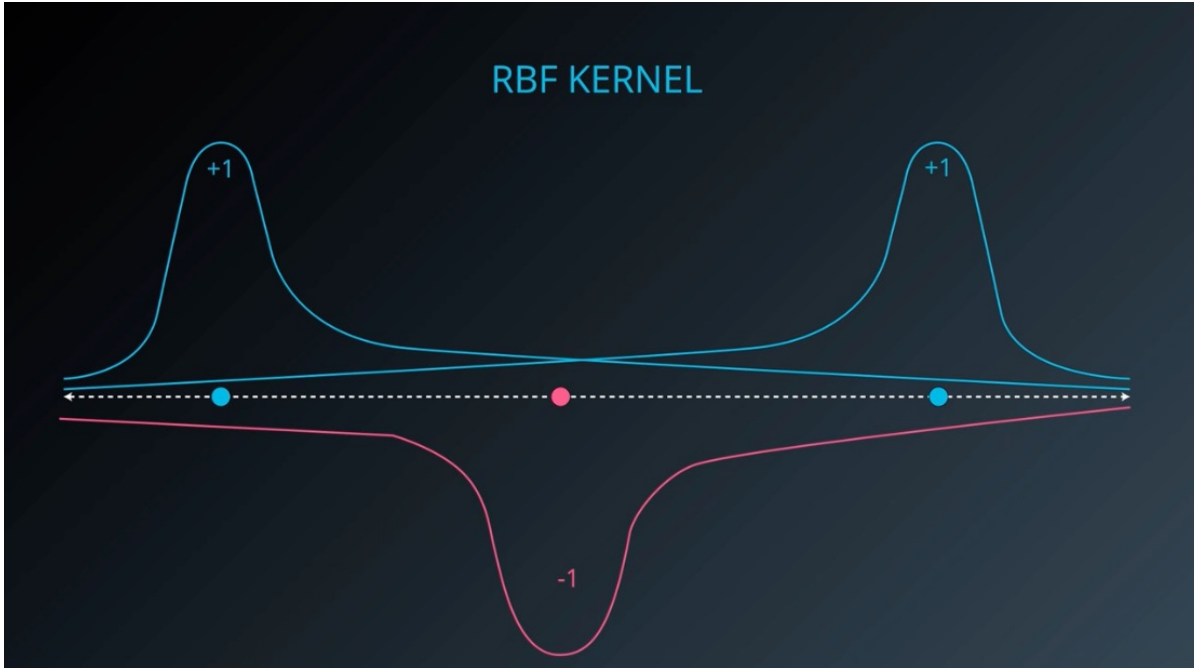
# Machines à vecteurs de support: Le noyau gaussien (RBF)

**Comment trouver la série de gaussiennes qui a permis d'identifier notre frontière de décision ?**



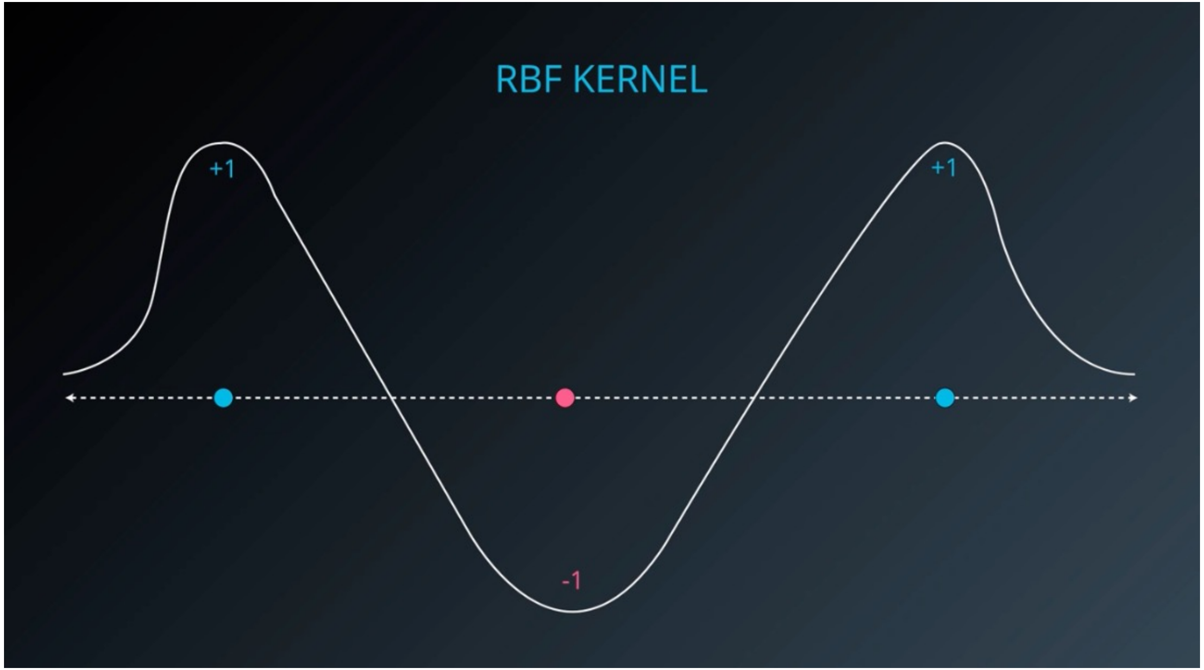

**Une gaussienne est tracée en face de chaque point à classer.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



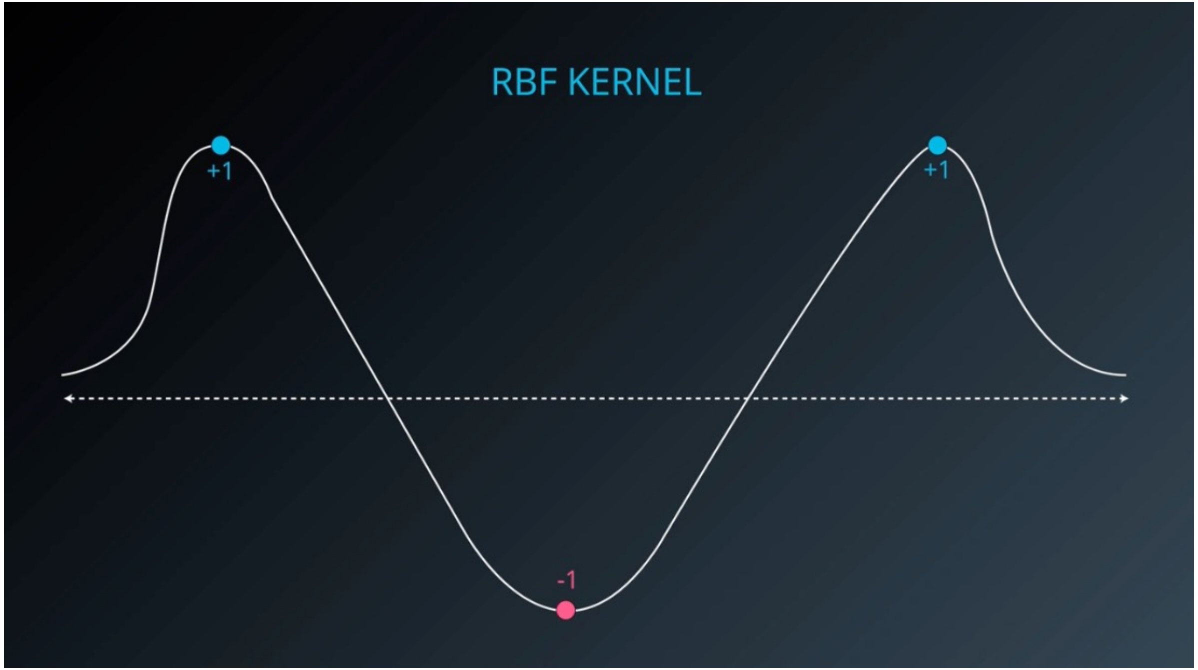

**Les gaussiennes correspondant aux points bleus sont multipliées par 1, tandis que celles des points rouges sont multipliées par -1.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



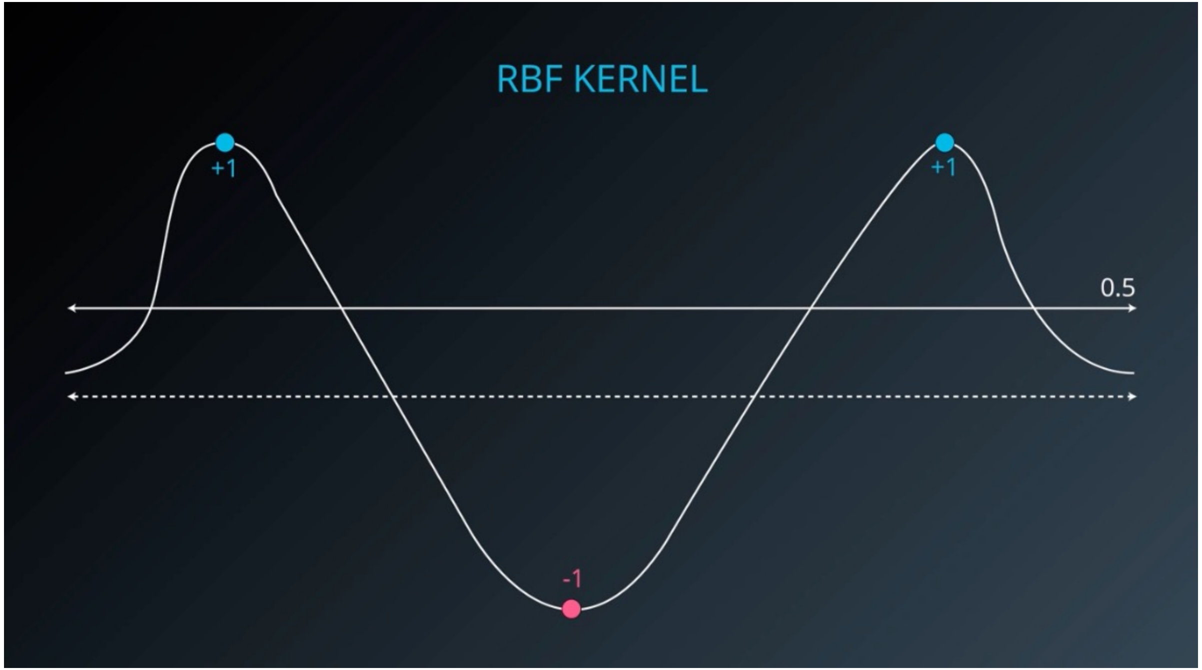

**Dans cet exemple, on additionne les trois gaussiennes.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



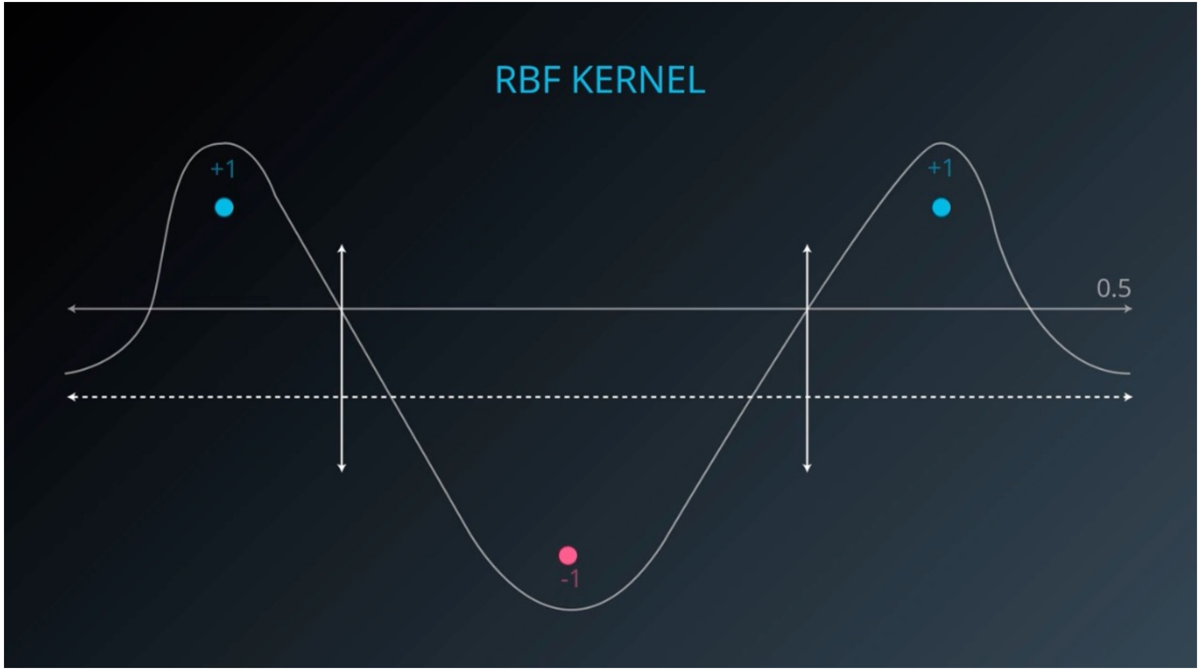

**Ensuite, les points à classer sont projetés sur le sommet de chaque gaussienne correspondante.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



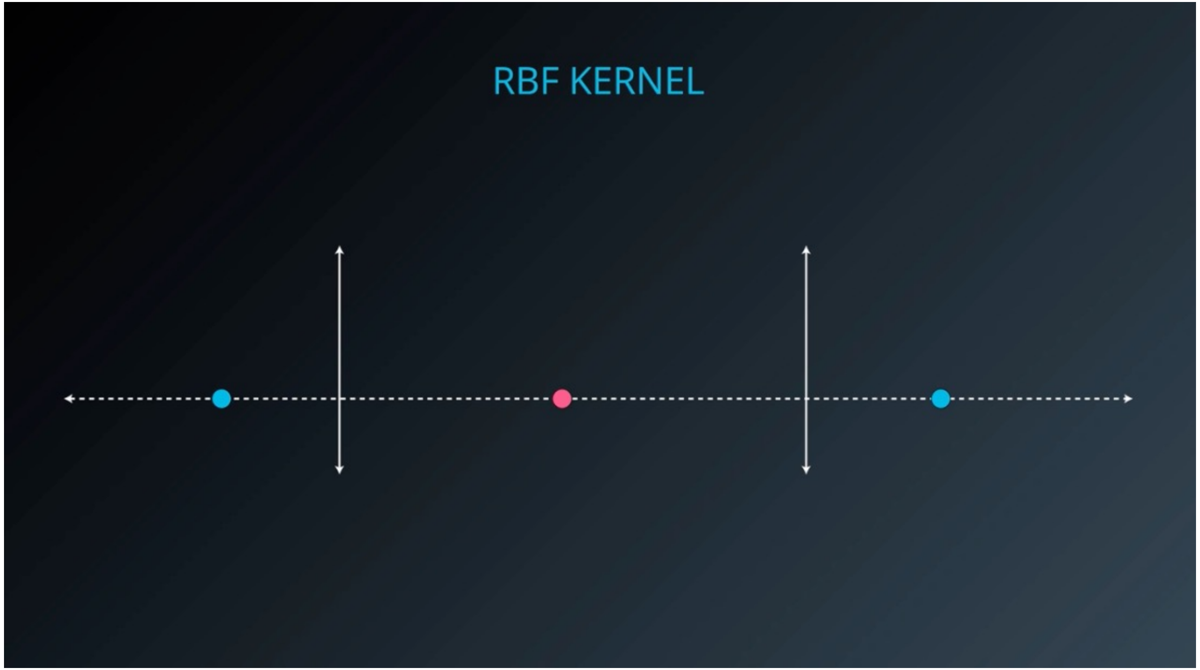

**Ensuite, une droite est tracée pour séparer les points bleus des points rouges.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)




**Ensuite, on détermine les points d'intersection entre la droite séparatrice et la série de gaussiennes.**

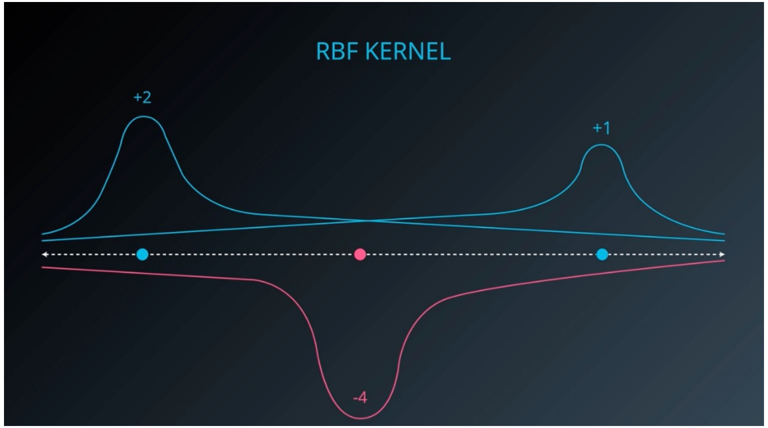
# Machines à vecteurs de support: Le noyau gaussien (RBF)

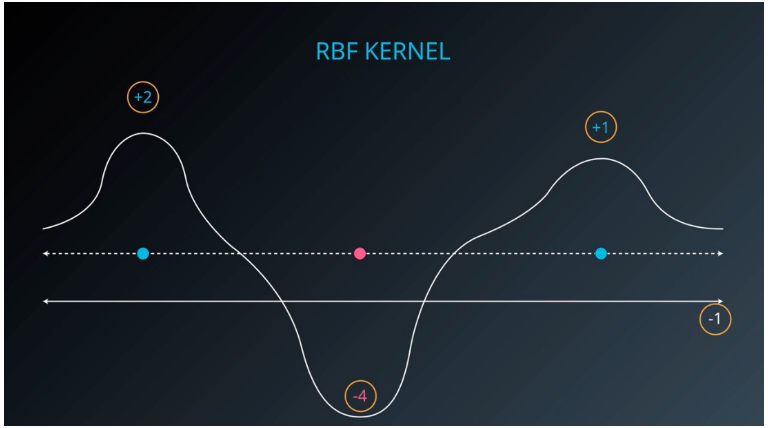



**Pour conclure, nous revenons à notre espace initial. Les points d'intersection que nous avons identifiés précédemment définissent la frontière de décision dans cet espace d'origine.**

**Cette frontière sépare les différentes classes que notre modèle cherche à classer.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)

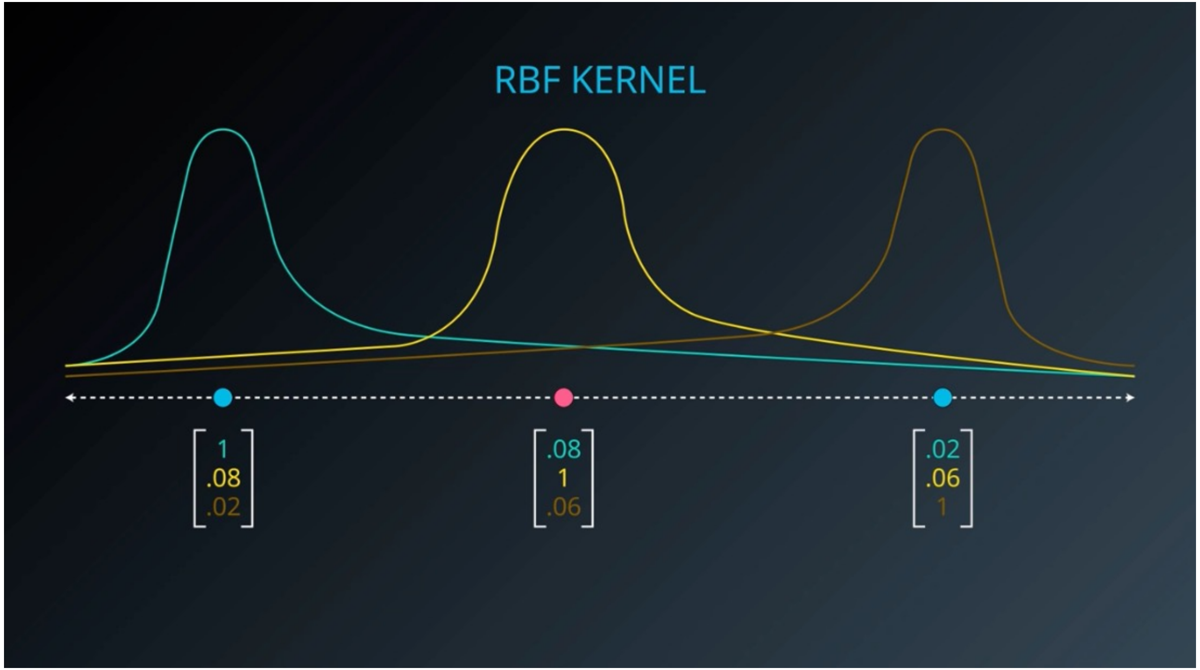



**En ajustant les coefficients multiplicateurs de notre série de gaussiennes, nous pouvons générer une multitude de modèles pour classifier nos données. Le défi consiste à identifier les coefficients optimaux qui maximisent la marge de séparation entre les classes, assurant ainsi une classification plus robuste et précise.**

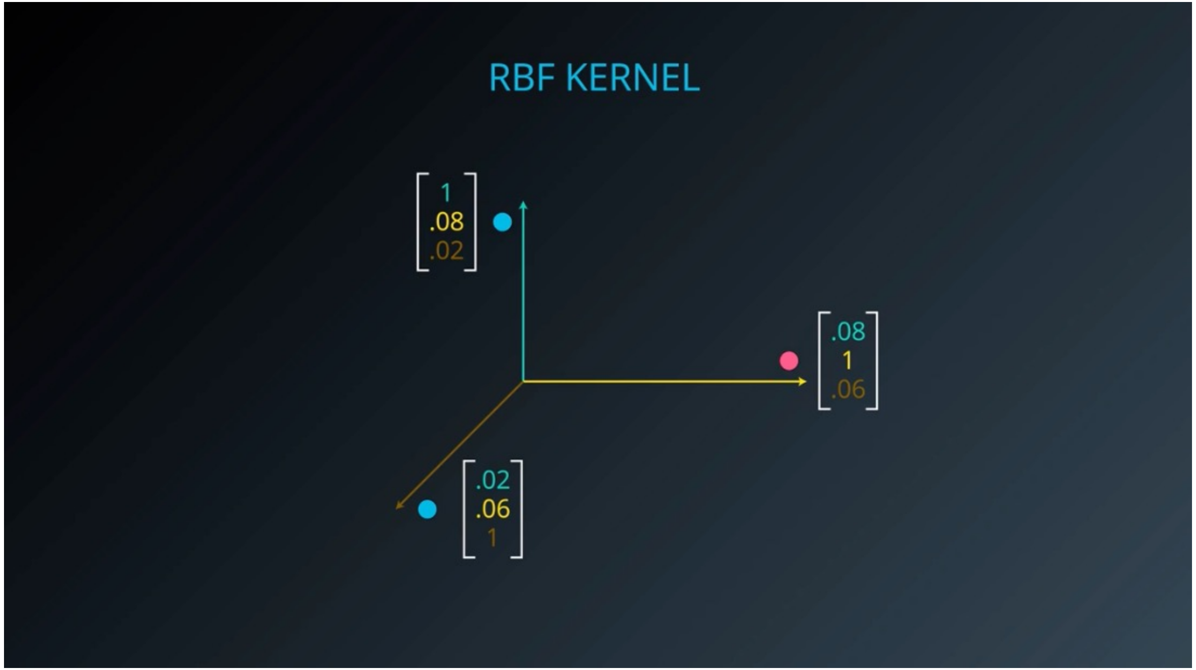
# Machines à vecteurs de support: Le noyau gaussien (RBF)

**Comment trouver les poids optimaux?**



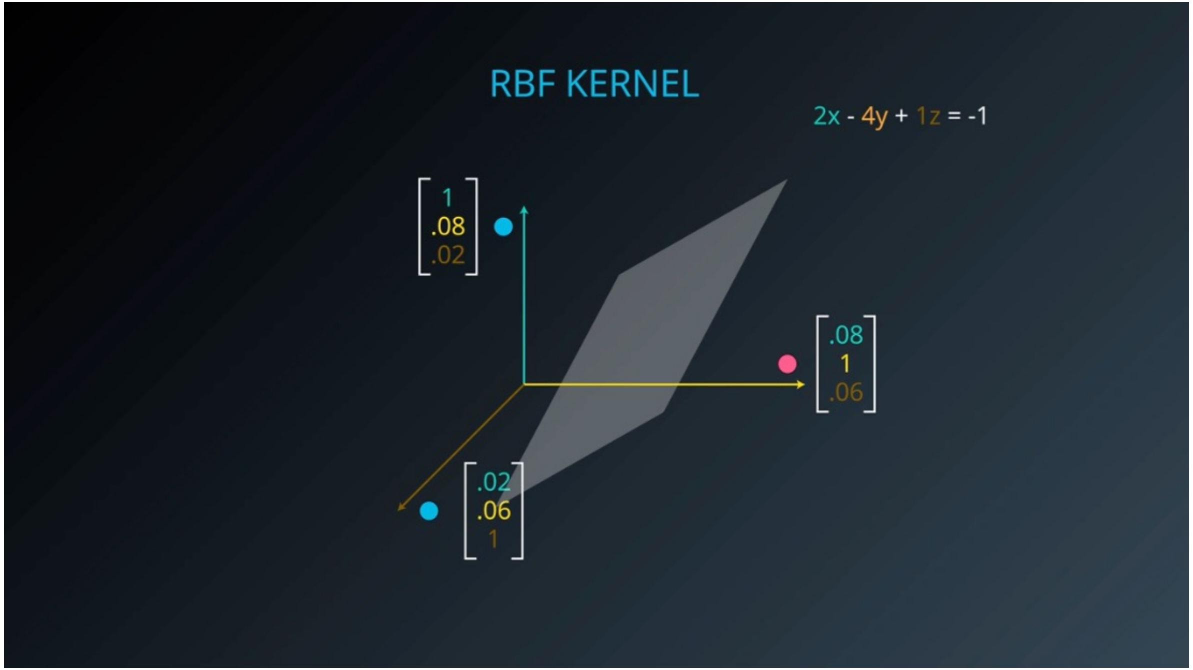

**Pour chaque gaussienne, on génère un vecteur contenant les valeurs (hauteurs) de toutes les gaussiennes disponibles à ce point spécifique. Ce vecteur représente ainsi le profil d'activation de l'ensemble des gaussiennes pour chaque position considérée.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



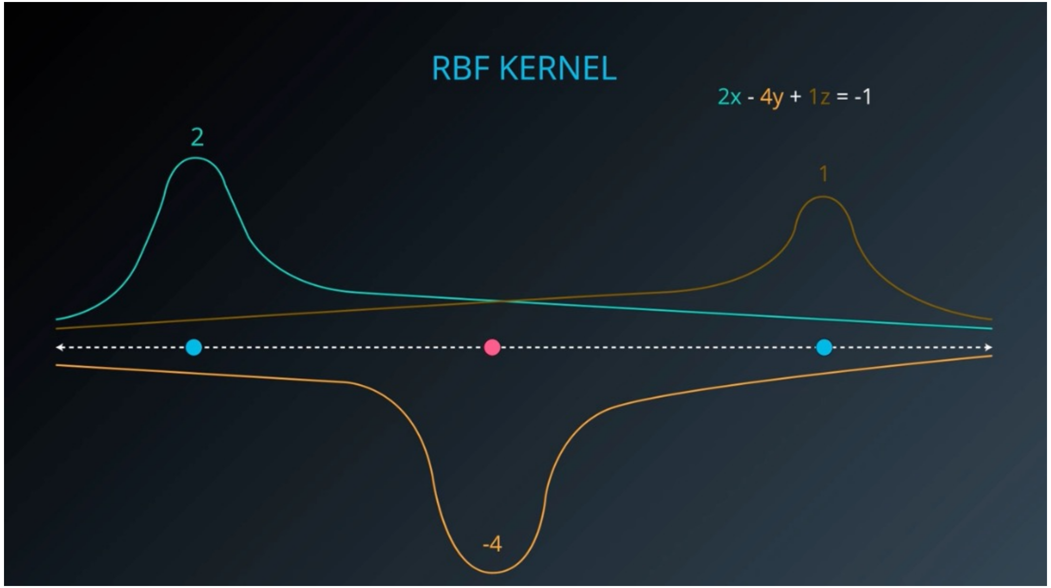

**Nous allons ensuite représenter graphiquement ces coordonnées dans un espace de dimension supérieure (dans notre exemple, un espace tridimensionnel). Cette projection dans un espace de plus haute dimension nous permettra de visualiser la séparation des données qui n'était pas linéairement possible dans l'espace d'origine.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



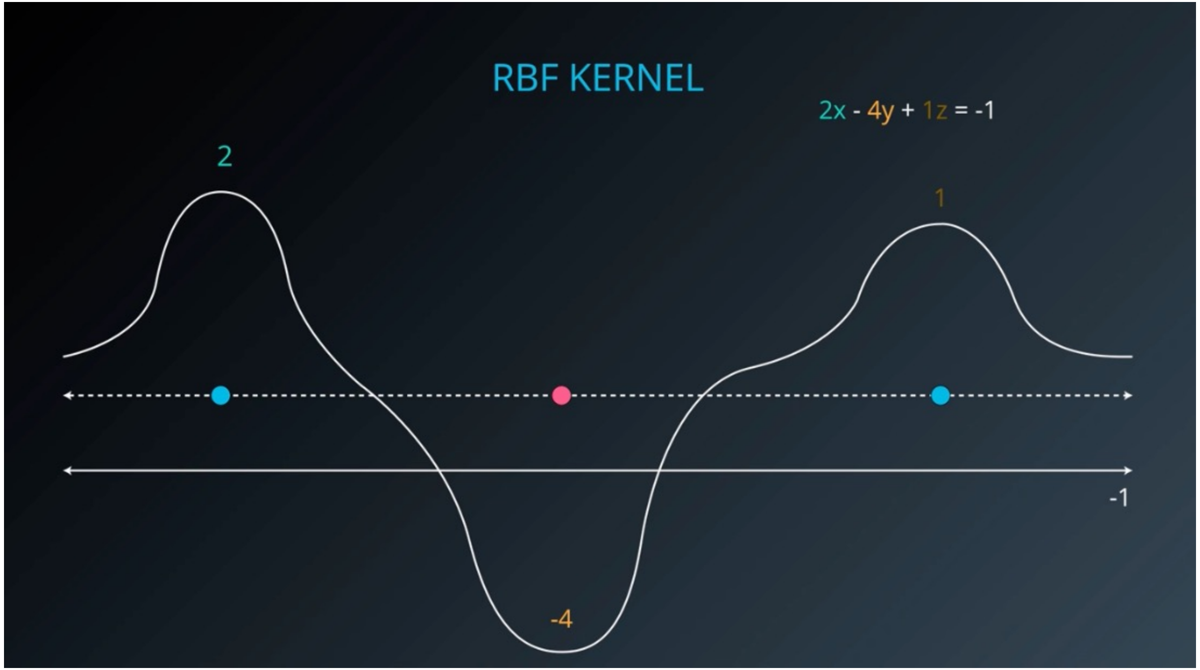

**En appliquant l'algorithme SVM (Support Vector Machine), nous déterminons l'équation de l'hyperplan optimal qui maximise la marge de séparation entre les différentes classes de points. Cet hyperplan constitue la frontière de décision la plus robuste pour notre classification.**

# Machines à vecteurs de support: Le noyau gaussien (RBF)



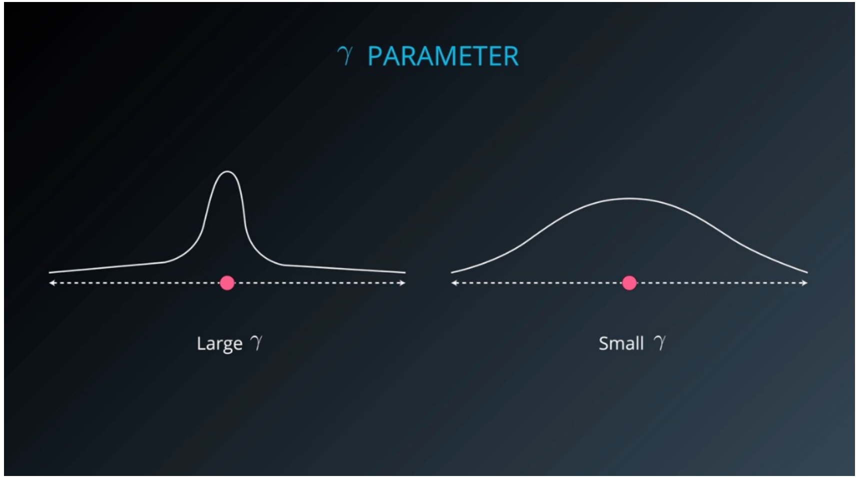

**Les coefficients obtenus pour l'équation de l'hyperplan optimal deviennent les poids de notre modèle. Ces poids sont utilisés pour multiplier chaque gaussienne dans notre série, définissant ainsi la combinaison linéaire qui forme notre frontière de décision finale dans l'espace d'origine.**

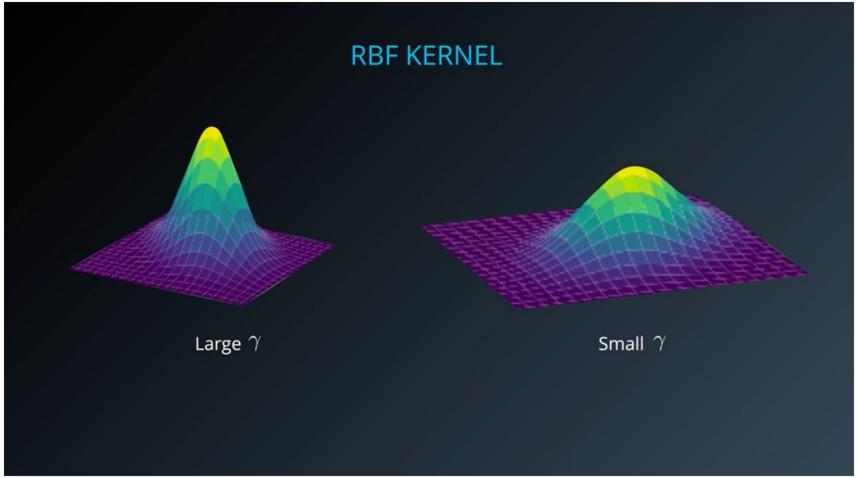
# Machines à vecteurs de support: Le noyau gaussien (RBF)




**Le terme constant de l'équation de notre hyperplan, dans ce cas -1, détermine le seuil de décision. Ce seuil définit la position exacte de la frontière séparatrice dans l'espace des caractéristiques, permettant de classifier les points de part et d'autre de cette frontière.**

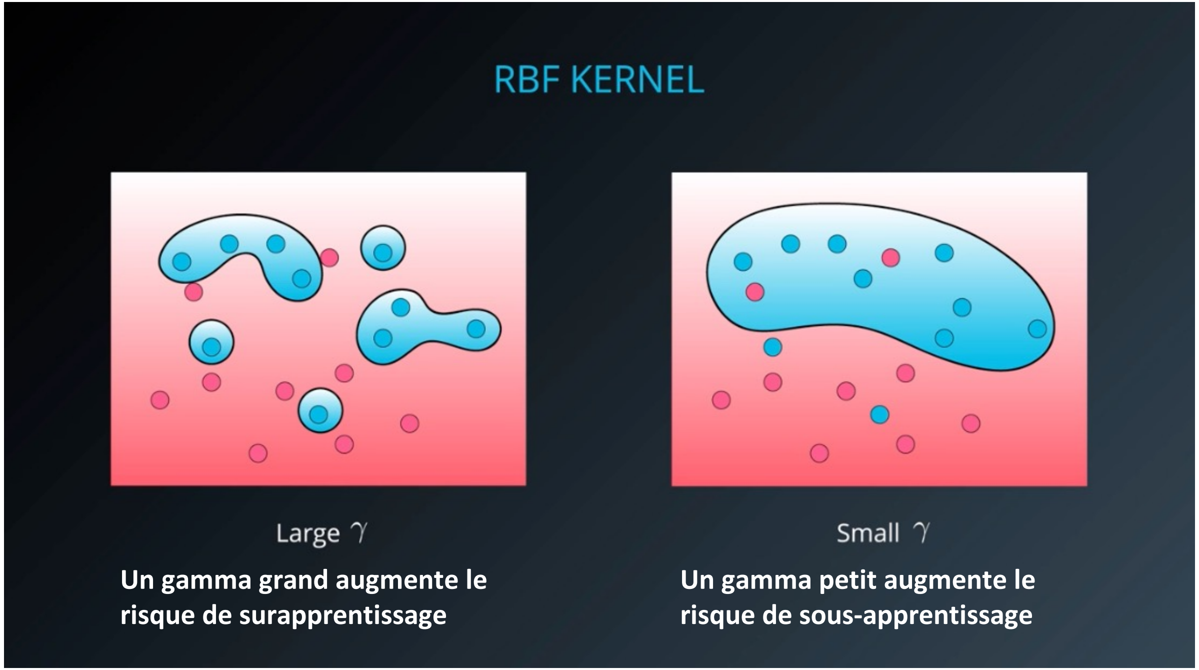
# Machines à vecteurs de support: Le paramètre $\gamma$



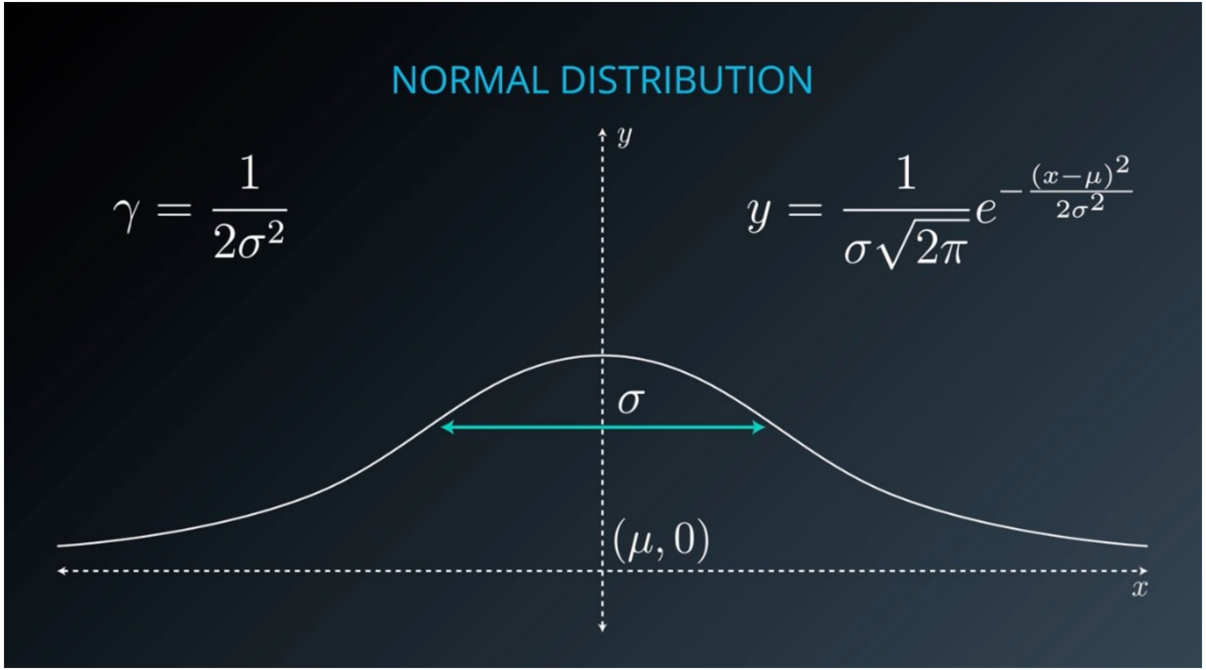




**Le paramètre gamma $\gamma$ détermine la forme des gaussiennes utilisées dans notre modèle. Sa valeur optimale est sélectionnée durant la phase d'apprentissage, en fonction des spécificités du problème à résoudre. Ce paramètre influence directement la flexibilité et la capacité de généralisation du modèle.**

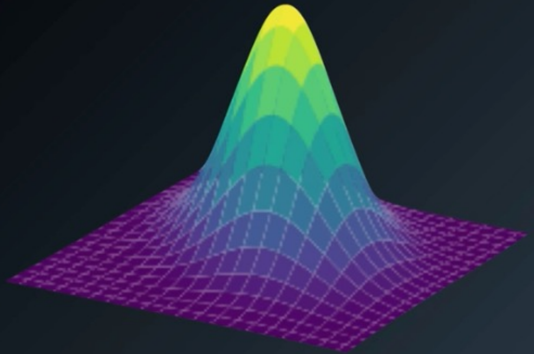
# Machines à vecteurs de support: Le paramètre $\gamma$



# Machines à vecteurs de support: Le paramètre $\gamma$



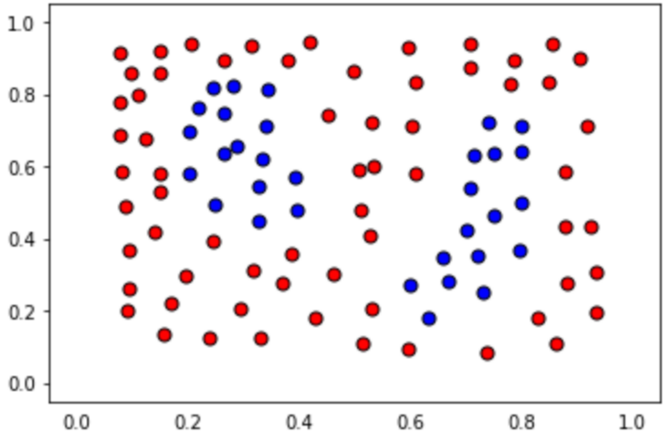
# Machines à vecteurs de support: Le paramètre $\gamma$




$$z = \frac{1}{2\pi\sigma_X\sigma_Y\sqrt{1-\rho^2}} exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_X)^2}{\sigma_X^2}+\frac{(y-\mu_Y)^2}{\sigma_Y^2}-\frac{2\rho(x-\mu_X)(y-\mu_Y)}{\sigma_X\sigma_Y}\right]\right)$$

# Machines à vecteurs de support : Exercice

Dans ce quiz, on vous fournira l'ensemble de données échantillon suivant, et votre objectif sera de définir un modèle qui atteint une précision de 100% sur celui-ci.




https://github.com/elhidali/EPISEN-2024