2022/2023

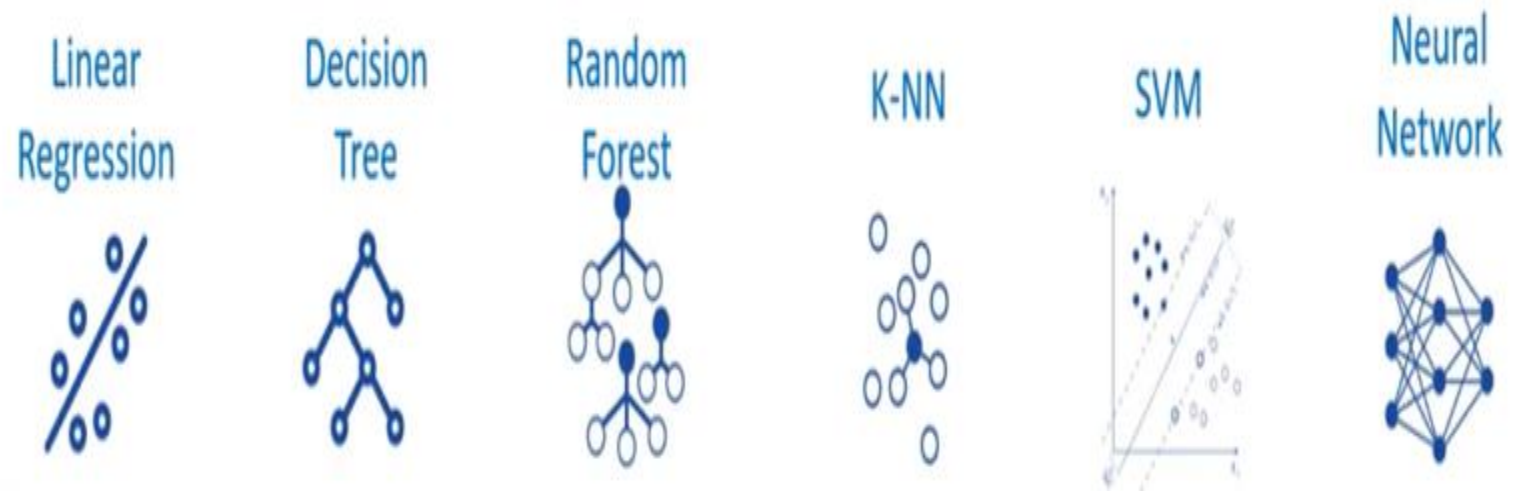
Support Vecteur Machines SVM

Réalisé par: Naïma Daghfous

Rappel

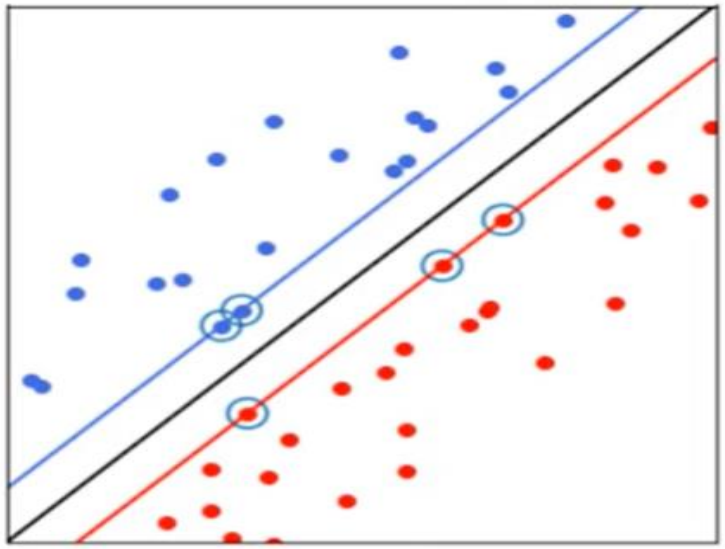
- Apprentissage automatique = apprendre un modèle formel à partir de données observées

- Le Machine Learning consiste à créer un **modèle** à l'aide de données pour permettre à un ordinateur d'apprendre à effectuer une tâche spécifique.



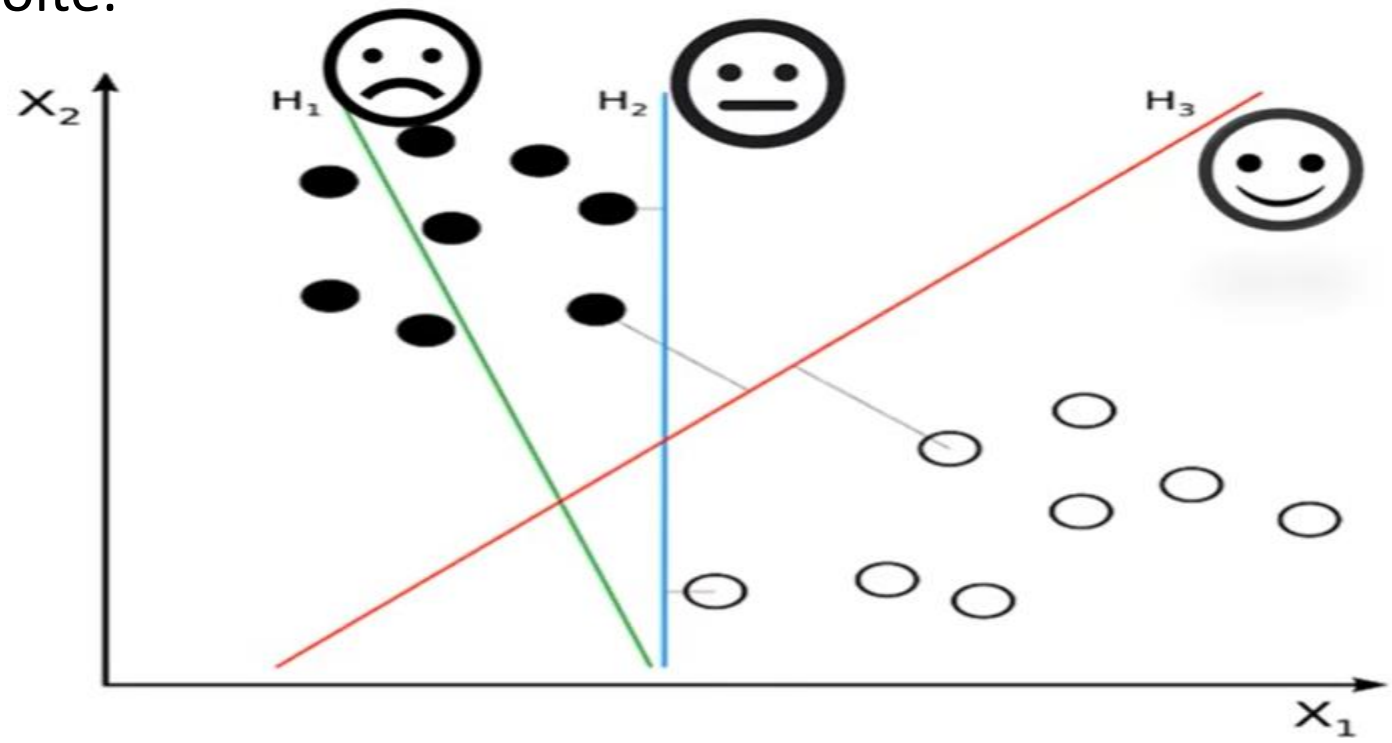
SVM

- Le SVM est un modèle d'apprentissage automatique supervisé qui est principalement utilisé pour les classifications (mais il peut aussi être utilisé pour la régression !). L'intuition derrière les **Support Vector Machines** est de simplement séparer des données en les délimitant (créer des frontières) afin de créer des groupes. Exemple :



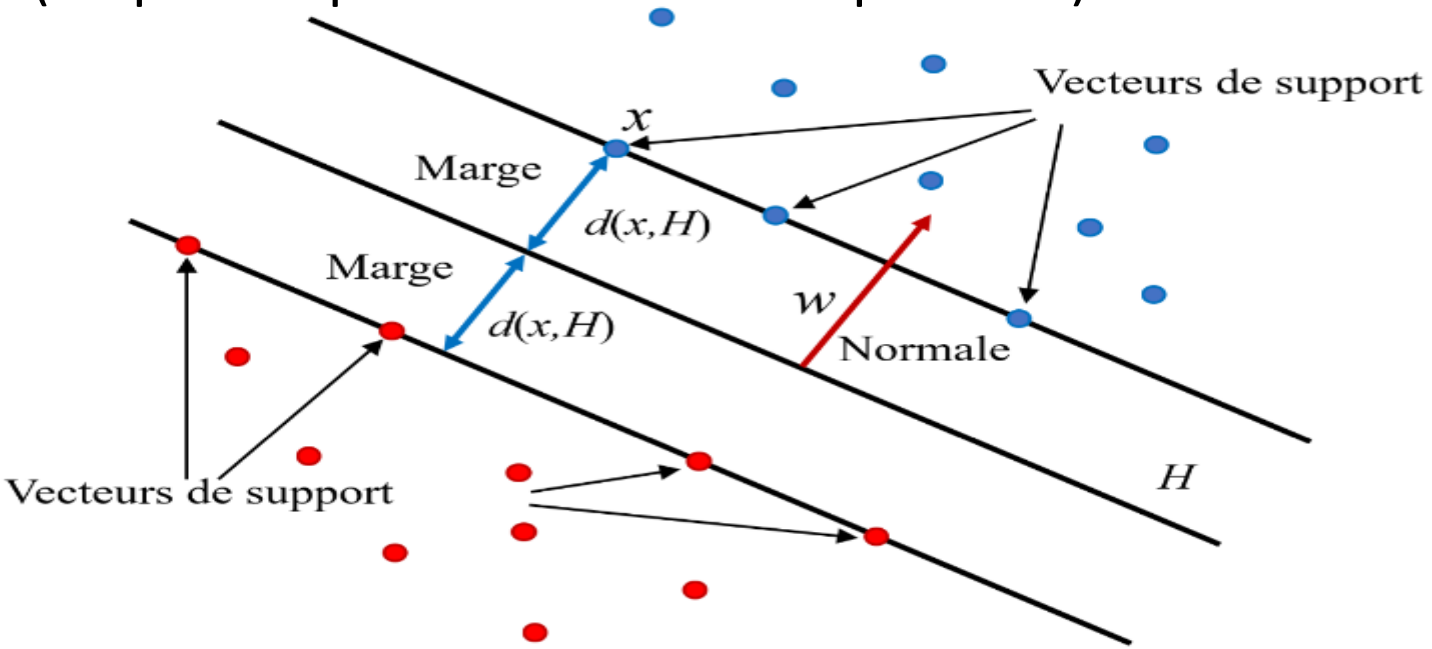
SVM

- Les SVM ont pour objectif de trouver, dans un espace de dimension N>1, l'hyperplan qui divise au mieux un jeu de donnée en deux. Les SVM sont des séparateurs linéaires, c'est-à-dire que la frontière séparant les classes est une droite.



Marge & Vecteurs de Support

- La frontière choisie doit maximiser sa distance avec les points les plus proches de la frontière(**marge**). Les points d'entraînement les plus proches de la frontière sont appelés **vecteurs support**. Ils sont appelés comme cela car la frontière donnée par un SVM ne dépend que des vecteurs support (on peut le prouver mathématiquement).



Marge & Vecteurs de Support

Rappel (pb classification à deux classes)

- $\mathcal{X}$ espace quelconque d'objets
- $\mathcal{Y} = \{-1, 1\}$ (classification).
- Données : On dispose d'un échantillon $S_n = (x_1, y_1), \ldots, (x_n, y_n)$
- But : Construire un classifieur $g : \mathcal{X} \to \{-1, 1\}$ à partir de S_n
- Plutôt que de construire directement g on construit $f : \mathcal{X} \to \mathbb{R}$
- Le classifieur est donné par le signe de f

$$g = \text{sgn}(f)$$

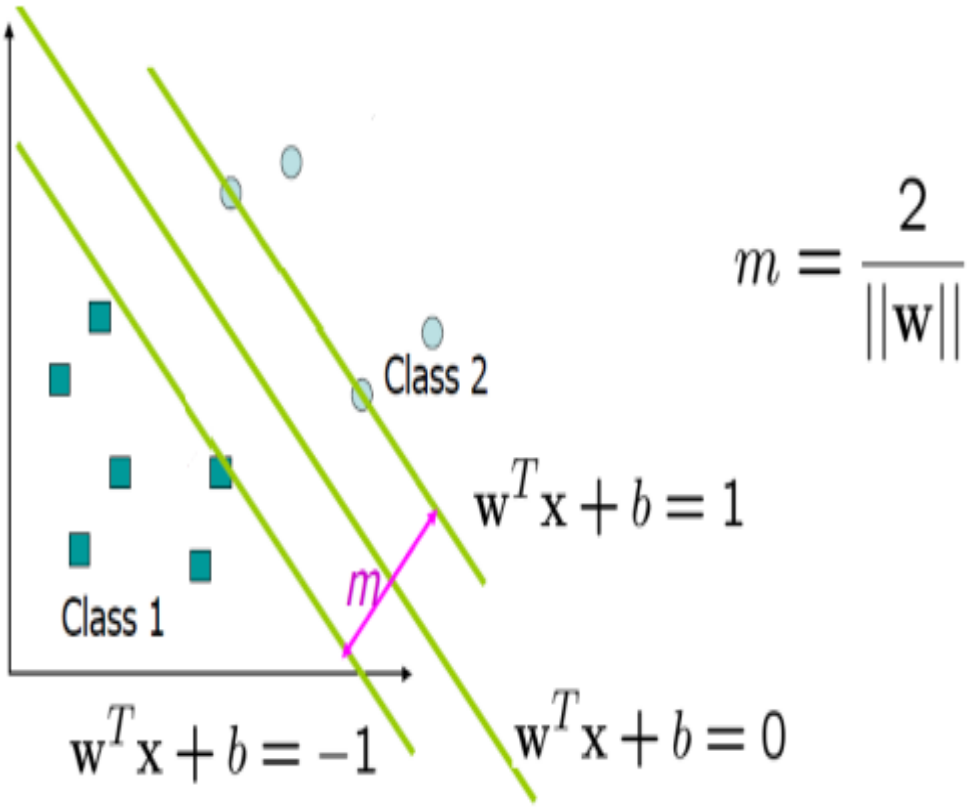
On suppose d'abord que les données d'apprentissage sont linéairement séparables, c'est à dire qu'il existe un hyperplan qui sépare les données sans erreur. Dans ce cas, on cherche **l'hyperplan de marge maximale** :

$$f(x) = \langle w, x \rangle + b = w^\top x + b$$

Avec w et b deux inconnus

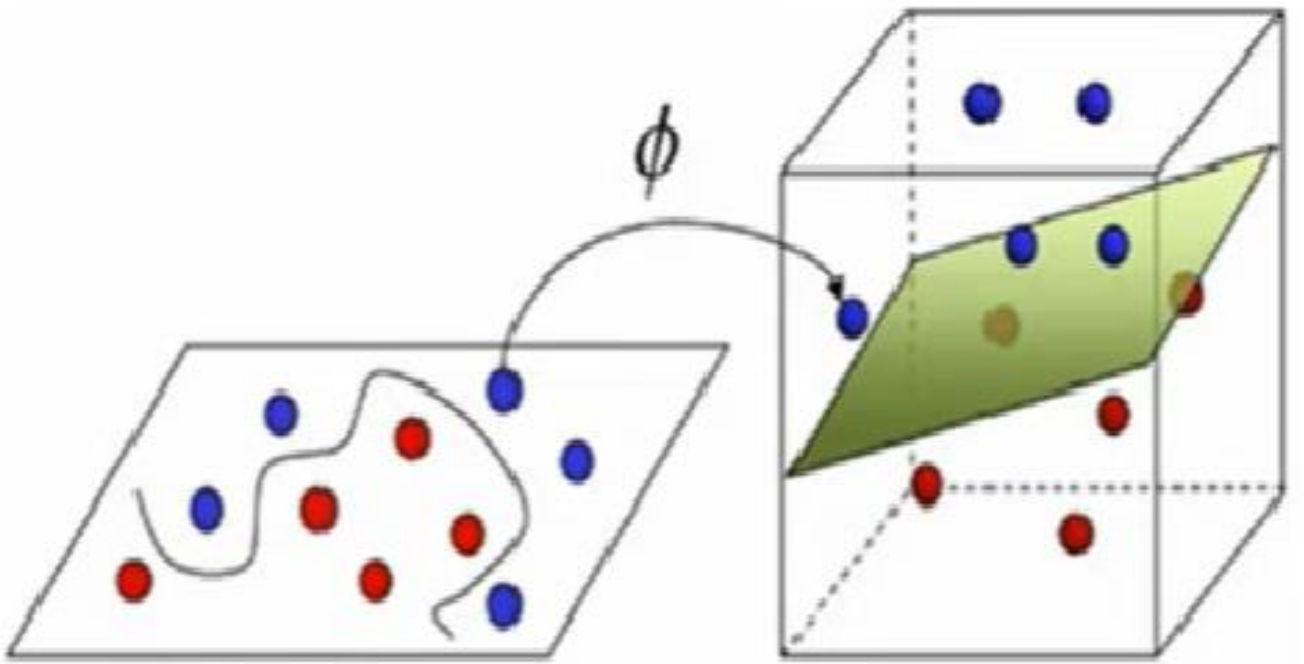
Marge & Vecteurs de Support

marge = distance du point le plus proche la l'hyperplan



$$m = \frac{2}{||w||}$$

Fonction Noyau



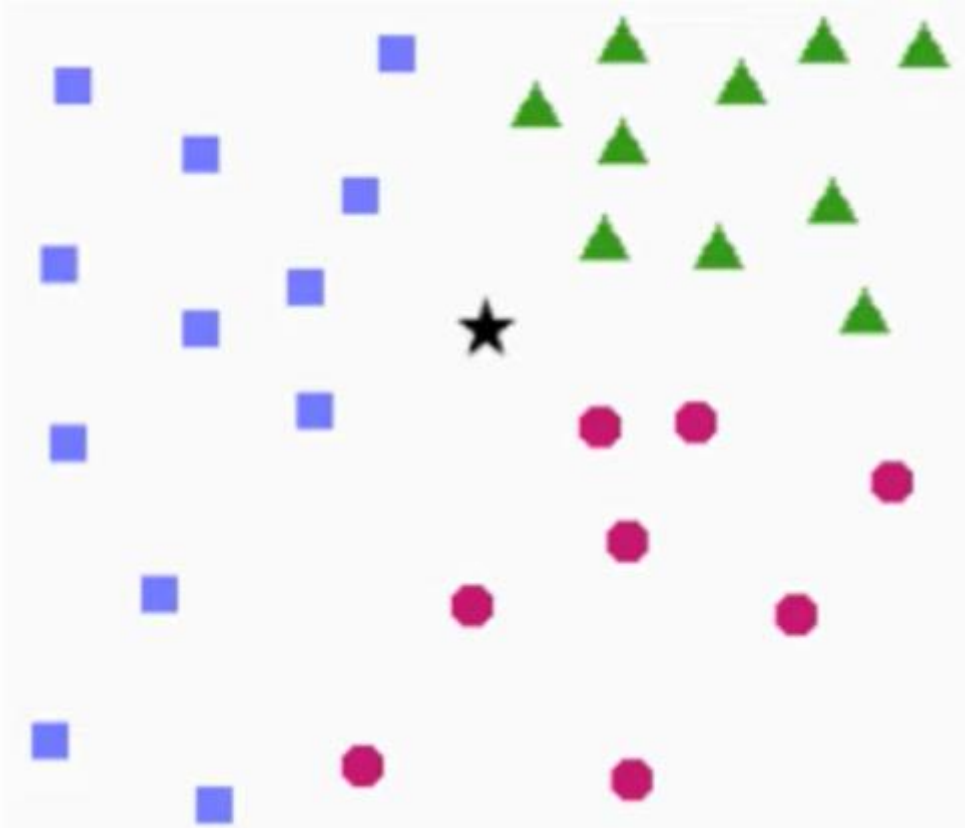
kernel trick (Astuce du noyau)

Fonction Noyau

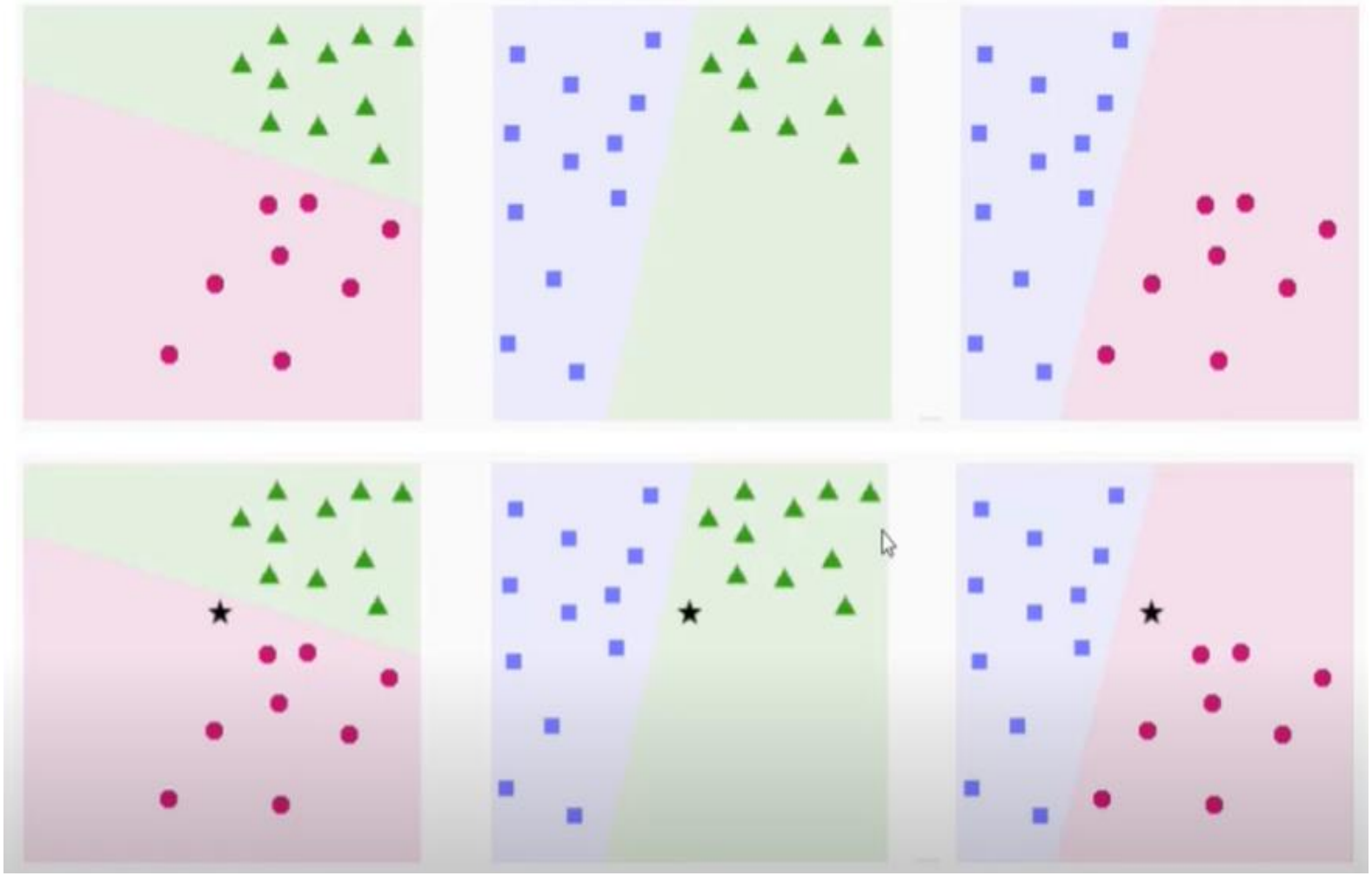
SVM utilisent différents types de fonctions du noyau. Ces fonctions peuvent être de différents types. Par exemple

- linéaire,
- non linéaire,
- polynomiale,
- fonction de base radiale (RBF)
- sigmoïde

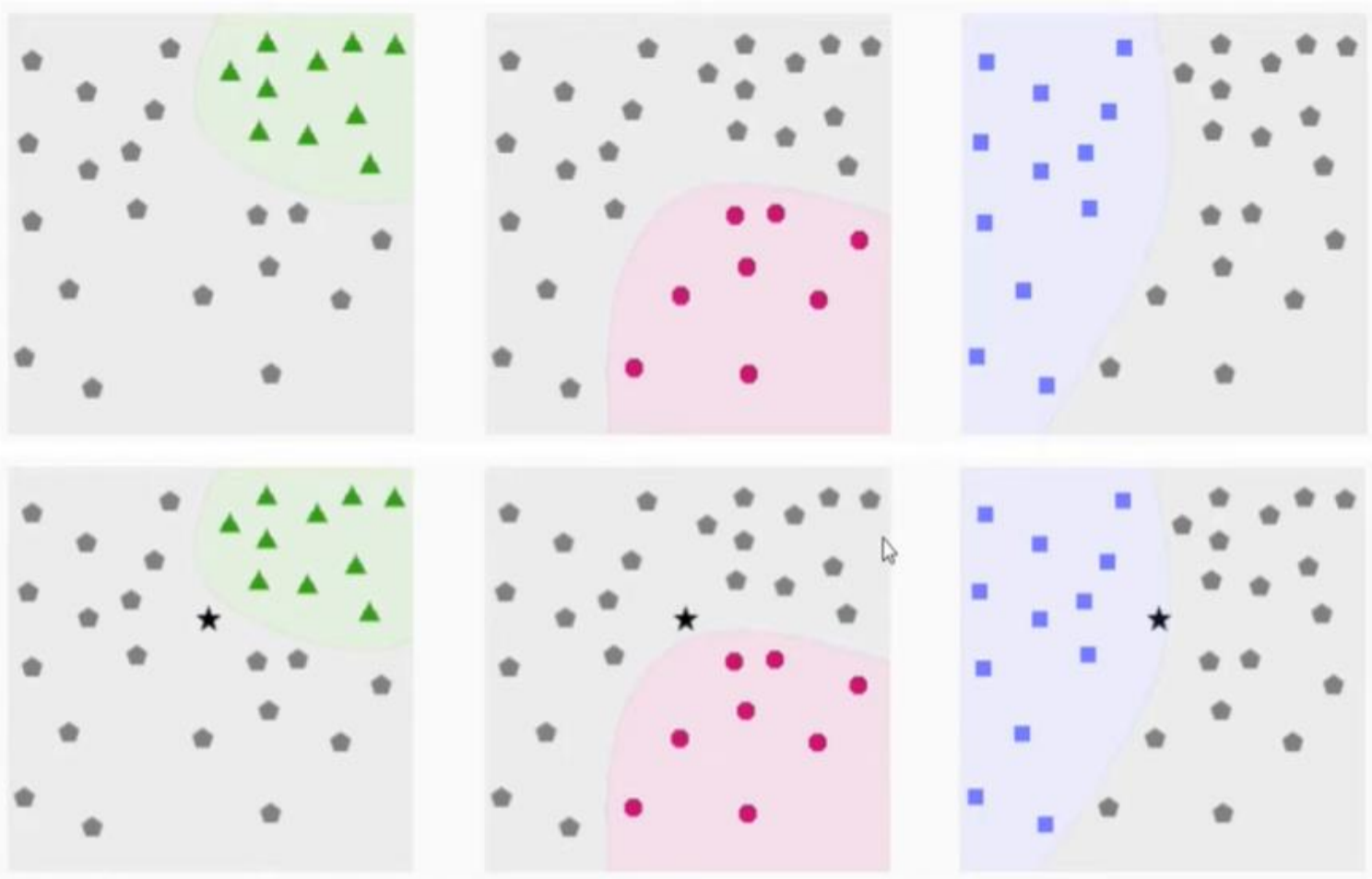
SVM Multi Classes



SVM Multi Classes : one vs one



SVM Multi Classes : one vs all



SVM: Pratique

Classification des billets

- Variance
- Skewness
- Curtosis
- Entropy

- Authentique (1)
- Non authentique (0)

SVM: Pratique

```
# Importer les librairies
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
```

```
# importer dataset
Data=pd.read_csv('bill_authentication.csv')
X=Data.iloc[:,:-1].values
y=Data.iloc[:,-1].values
```

```
#Aperçu des données
Data.head()
```

	Variance	Skewness	Curtosis	Entropy	Class
0	3.62160	8.6661	-2.8073	-0.44699	0
1	4.54590	8.1674	-2.4586	-1.46210	0
2	3.86600	-2.6383	1.9242	0.10645	0
3	3.45660	9.5228	-4.0112	-3.59440	0
4	0.32924	-4.4552	4.5718	-0.98880	0

SVM: Pratique

```
#Diviser les données en jeu d'entrainement et jeu de test
X_train, X_test, y_train, y_test = \
    train_test_split(X,y,test_size=0.2)
```

```
#Créer le modèle SVM
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
```

```
SVC(kernel='linear', random_state=0)
```

```
#Prediction sur le Test set
y_pred = classifier.predict(X_test)
```

```
#Matrice de confusion
confusion_matrix(y_test, y_pred)
```

```
array([[138,   4],
       [  1, 132]], dtype=int64)
```

SVM: Pratique

```
#Rapport de classification
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      0.97      0.98       142
           1       0.97      0.99      0.98       133

    accuracy                           0.98       275
   macro avg       0.98      0.98      0.98       275
weighted avg       0.98      0.98      0.98       275
```