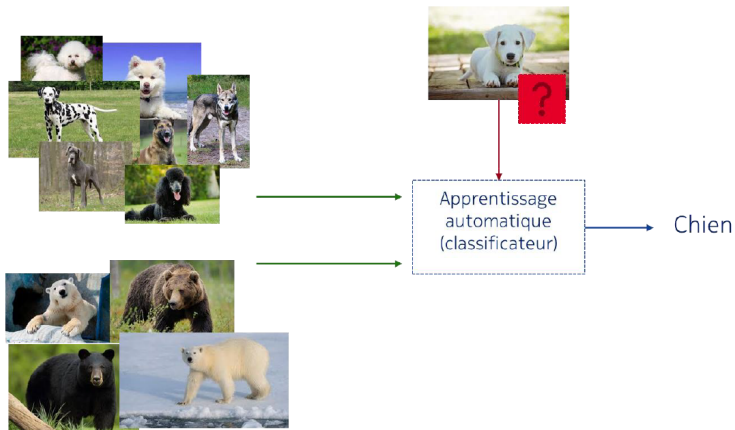


Apprentissage automatique supervisé

Plan du cours

- ▶ Classification
- ▶ Régression linéaire
- ▶ Quelques algo

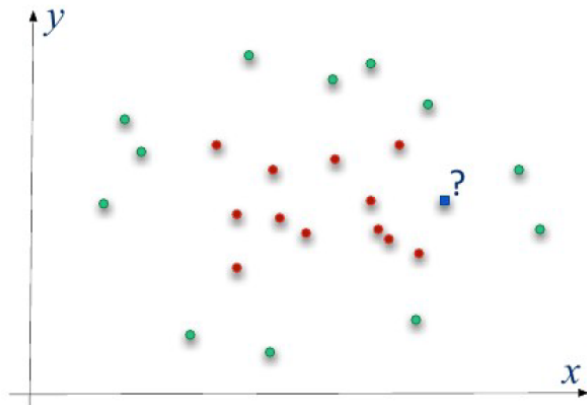
Classification : Rappel



Classification : Rappel

On dispose d'exemples d'apprentissage : 1 point dans l'espace,
une couleur, une classe...

On souhaite : déterminer la couleur, la classe d'un exemple nouveau.
Les valeurs sont discrètes.

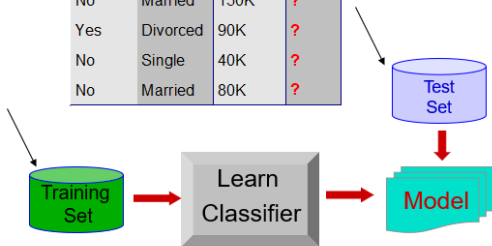


Classification : Exemple

categorique
categorique
continue
classe

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Refund	Marital Status	Taxable Income	Cheat
No	Single	75K	?
Yes	Married	50K	?
No	Married	150K	?
Yes	Divorced	90K	?
No	Single	40K	?
No	Married	80K	?



Classification : Application

- Reconnaissance des formes
 - ▶ Reconnaissance de visages : reconnaître les personnes malgré les variations (pose, éclairage, lunettes, maquillage, coiffure)
 - ▶ Reconnaissance de caractères manuscrits : reconnaître malgré différents styles d'écriture
 - ▶ Reconnaissance de la parole : dépendance temporelle de l'information, utiliser des dictionnaires de mots/structures valides
- Aide au diagnostic médical : déterminer les problèmes médicaux à partir des symptômes
- Extraction de connaissances : expliquer par des règles simples des masses de données
- Compression : représenter par des modèles compacts les données
- Détection d'irrégularités : identifier des fraudes, des intrusions

Classification : Exemple

Détection de fraudes :

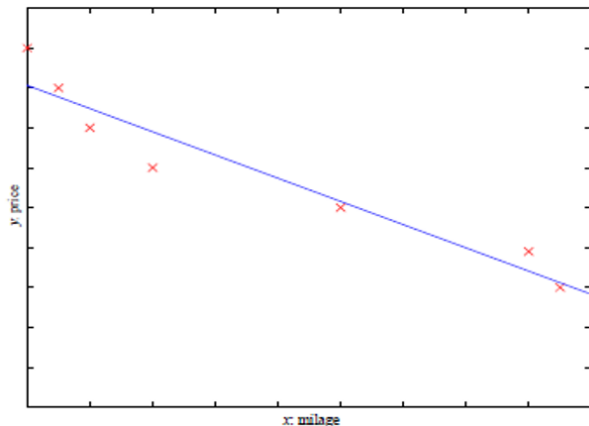
- ▶ But : Détecter les transactions frauduleuses des CB
- ▶ Approche :
 - ▶ Attributs : les transactions passées + informations du titulaire
 - ▶ Classes : labéliser les transactions passées comme correcte ou fraude
 - ▶ Apprendre un modèle de la classe transaction
 - ▶ Utiliser ce modèle pour détecter les fraudes en observant les transactions d'une CB d'un compte donné

Régression : Rappel

On dispose d'exemples d'apprentissage : 1 point dans l'espace,
une valeur quantitative

On souhaite : la valeur associée à un exemple nouveau

Les valeurs sont continues.



Régression : Exemple

Y est la valeur réelle à prédire et $h()$ est la fonction de régression.

- ▶ Prédiction du prix d'une voiture à partir du km parcouru :
 - ▶ Observations : le km parcouru (X)
 - ▶ Prédiction : le prix (Y)
- ▶ Applications en prédiction de valeurs :
 - ▶ Économie
 - ▶ Météorologie

Quelques classifieurs

- ▶ Classification Naive Bayesienne
- ▶ Arbres de décision
- ▶ KNN (plus proche voisin)
- ▶ SVM (machines à vecteurs de support)

Classification Naive Bayesienne

La classification est basée sur le théorème de Bayes.

Le théorème de Bayes est l'un des principaux théorèmes de la théorie des probabilités,

Déterminer la probabilité qu'un événement arrive à partir d'un autre événement qui s'est réalisé,

à condition que ces deux événements soient interdépendants.

- ▶ `from sklearn.naive_bayes import GaussianNB`

Classification Naive Bayesienne

Une approche probabiliste pour la résolution de problèmes de

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}, \text{ à condition que } P(B) \neq 0$$

classification.

A et B sont deux évènements,

$P(A)$ et $P(B)$ sont la probabilité des deux évènements

$P(A| B)$ est la probabilité que A se réalise sachant B.

$P(B| A)$ est la probabilité que B se réalise sachant A.

Théorème de Bayes : Exemple

Étant donné :

- ▶ Un médecin sait que la méningite cause la raideur de la nuque dans 50% des cas
- ▶ la probabilité à priori qu'un patient souffre de la méningite est de $1 / 50000$
- ▶ la probabilité à priori qu'un patient ait une raideur de la nuque est de $1 / 20$

Si un patient a une raideur de la nuque, quelle est la probabilité qu'il ait la méningite?

Théorème de Bayes : Exemple

Étant donné :

- ▶ Un médecin sait que la méningite cause la raideur de la nuque dans 50% des cas
- ▶ la probabilité à priori qu'un patient souffre de la méningite est de $1 / 50000$
- ▶ la probabilité à priori qu'un patient ait une raideur de la nuque est de $1 / 20$

Si un patient a une raideur de la nuque, quelle est la probabilité qu'il ait la méningite?

$$P(M | R) = \frac{P(R | M)P(M)}{P(R)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$

Classifieur Bayésien

- ▶ Considérer chaque attribut et étiquette de classe comme des variables aléatoires
- ▶ Étant donné un enregistrement avec des attributs :
 (A_1, A_2, \dots, A_n)
- ▶ L'objectif est de prédire la classe C
- ▶ Plus précisément, nous voulons trouver la valeur de C qui maximise $P(C \mid A_1, A_2, \dots, A_n)$

Classifieur Bayésien : Exemple

Approche :

- ▶ Calculer la probabilité à-posteriori $P(C|A_1, A_2, \dots, A_n)$ pour toutes les valeurs de C en utilisant le théorème de Bayes.
- ▶ Choisir la valeur de C qui maximise :
 $P(C|A_1, A_2, \dots, A_n)$
- ▶ Equivalent à choisir la valeur de C qui maximise
 $P(A_1, A_2, \dots, A_n|C)P(C)$
- ▶ Comment estimer $P(A_1, A_2, \dots, A_n|C)$?

Classifieur Bayésien : Exemple

Supposons une indépendance entre les attributs A_i pour une classe donnée :

$$P(A_1, A_2, \dots, A_n | C_j) = P(A_1 | C_j) P(A_2 | C_j) \dots P(A_n | C_j)$$

Classifieur Bayésien : Exemple

Name	Give Birth	Can Fly	Live in Water	Have Legs	Class
human	yes	no	no	yes	mammals
python	no	no	no	no	non-mammals
salmon	no	no	yes	no	non-mammals
whale	yes	no	yes	no	mammals
frog	no	no	sometimes	yes	non-mammals
komodo	no	no	no	yes	non-mammals
bat	yes	yes	no	yes	mammals
pigeon	no	yes	no	yes	non-mammals
cat	yes	no	no	yes	mammals
leopard shark	yes	no	yes	no	non-mammals
turtle	no	no	sometimes	yes	non-mammals
penguin	no	no	sometimes	yes	non-mammals
porcupine	yes	no	no	yes	mammals
eel	no	no	yes	no	non-mammals
salamander	no	no	sometimes	yes	non-mammals
gila monster	no	no	no	yes	non-mammals
platypus	no	no	no	yes	mammals
owl	no	yes	no	yes	non-mammals
dolphin	yes	no	yes	no	mammals
eagle	no	yes	no	yes	non-mammals

Give Birth	Can Fly	Live in Water	Have Legs	Class
yes	no	yes	no	?

A: attributs

M: mammifères

N: non-mammifères

$$P(A | M) = \frac{6}{7} \times \frac{6}{7} \times \frac{2}{7} \times \frac{2}{7} = 0.06$$

$$P(A | N) = \frac{1}{13} \times \frac{10}{13} \times \frac{3}{13} \times \frac{4}{13} = 0.0042$$

$$P(A | M)P(M) = 0.06 \times \frac{7}{20} = 0.021$$

$$P(A | N)P(N) = 0.0042 \times \frac{13}{20} = 0.0027$$

$P(A|M)P(M) > P(A|N)P(N)$

=> Mammifère

Classifieur Bayésien : Avantages X Limites

- ▶ Algorithme de classification le plus simple et le plus rapide
- ▶ Ce modèle d'apprentissage ne nécessite qu'un nombre faible d'échantillons d'entraînement
- ▶ Il convient aussi aux larges ensembles de données.
- ▶ Robuste par rapport aux attributs non pertinents

- ▶ Suppose que les variables sont indépendantes.

Decision Trees

Un arbre de décision est composé de :

- Un nœud racine par lequel les données entrent.
- Des nœuds de décision représentant un classement de questions et de réponses qui conditionnent la question suivante.
- Des branches qui connectent entre les nœuds et pouvant être orientées.
- Dans ce cas l'un des nœuds de la connexion est appelé nœud parent et l'autre nœud enfant.

- ▶ `from sklearn.tree import DecisionTreeClassifier`
- ▶ `from sklearn.tree import DecisionTreeRegressor`

Decision Trees : Applications

- ▶ La volonté d'un client à acheter un produit donné.
- ▶ La planification des produits.
- ▶ Prise de décision générale pour les affaires.
- ▶ Approbation de prêt.
- ▶ Fraude aux impôts.

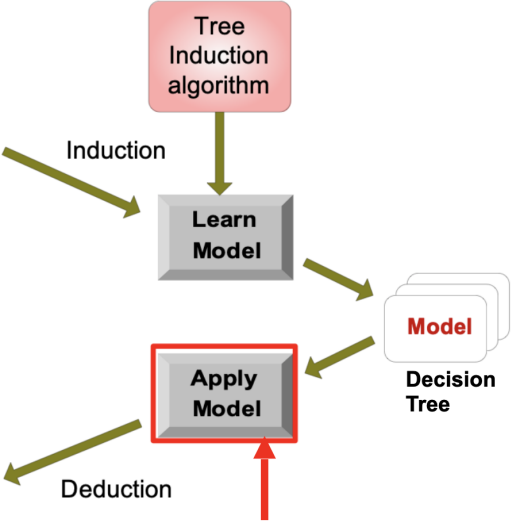
Decision Trees : Exemple

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set

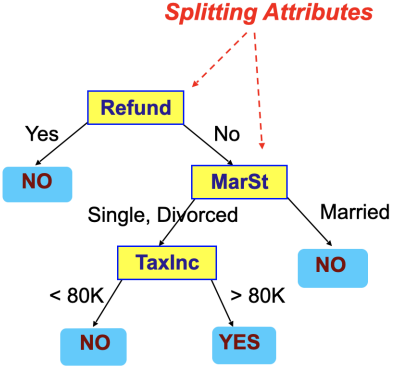


Decision Trees : Exemple

categorical
categorical
continuous
class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Données d'apprentissage

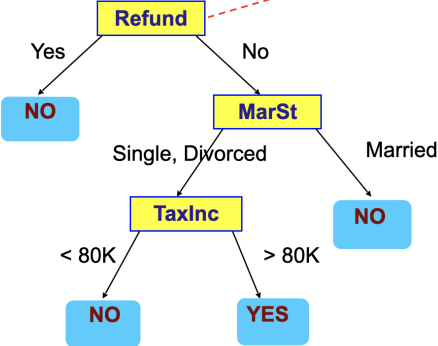


Modèle: Arbre de Décision

Decision Trees : Exemple

Données de Tests

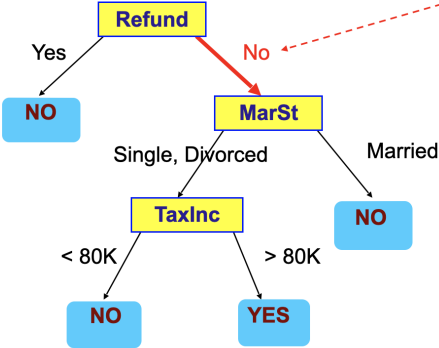
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Decision Trees : Exemple

Test Data

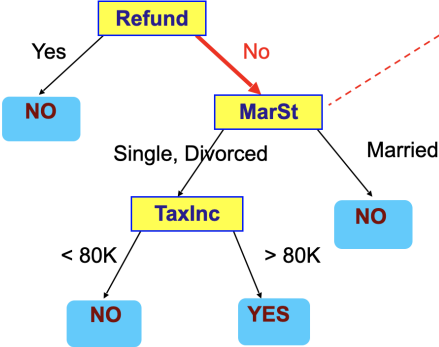
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



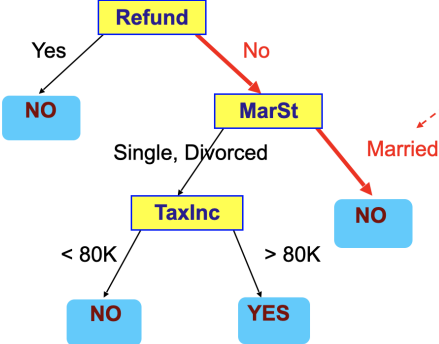
Decision Trees : Exemple

Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

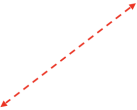


Decision Trees : Exemple

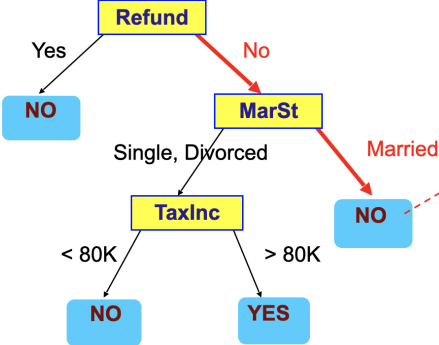


Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?



Decision Trees : Exemple



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assigner la valeur "No" à Cheat

Decision Trees : Avantages X Limites

- ▶ simplicité à comprendre et à visualiser
- ▶ faible nécessité de préparation de données.
- ▶ utiliser des données catégorielles ainsi que numériques.

- ▶ Les modèles d'arbre de décision peuvent quelque fois montrer du sur-apprentissage, c'est-à-dire que dans certains cas d'utilisations, les arbres générés sont trop complexes.
- ▶ Gourmandes

KNN (plus proche voisin)

Utiliser les k points les "plus proches" pour effectuer la classification.

Principe :

- ▶ calcule la distance et stocke les k voisins plus proches pour chaque échantillon de l'ensemble d'apprentissages
- ▶ pour un échantillon sans étiquette l'algorithme récupère les k voisins plus proches de celui-ci et prédit sa classe par vote.

Commandes :

- ▶ `from sklearn.neighbors import KNeighborsClassifier`
- ▶ `from sklearn.neighbors import KNeighborsRegressor`

KNN (plus proche voisin)

Set of Stored Cases

Atr1	AtrN	Class
			A
			B
			B
			C
			A
			C
			B

Ensemble d'exemples sauvegardés

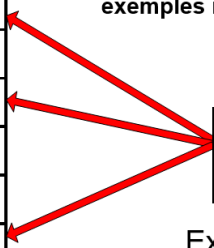
•Conserver les exemples d'apprentissage

•Utiliser des exemples d'apprentissage afin de prédire les étiquettes de classes des exemples non observés

Unseen Case

Atr1	AtrN

Exemple non observé



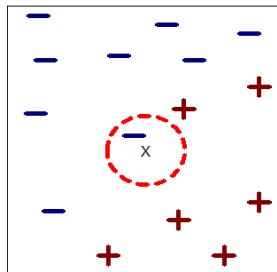
KNN : Applications

- ▶ La prédiction des prix des actions.
- ▶ Les systèmes de recommandation.
- ▶ L'analyse du risque de crédit.

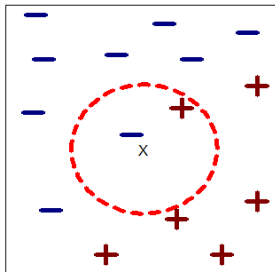
KNN : Principe

- ▶ Exigences :
 - ▶ L'ensemble des enregistrements
 - ▶ La mesure de distance (Distance Euclidienne,...)
 - ▶ La valeur de K
- ▶ Pour classer une instance :
 - ▶ Calculer la distance aux autres instances d'apprentissage
 - ▶ Identifier les k plus proches voisins
 - ▶ Utiliser les étiquettes de classe des K plus proches voisins (par exemple : par vote)

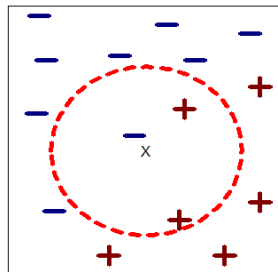
KNN : Principe



(a) 1-nearest neighbor
1-voisin-plus-proche



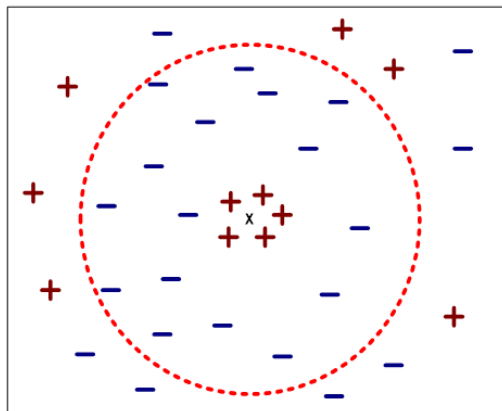
(b) 2-nearest neighbor
2-voisins-plus-proches



(c) 3-nearest neighbor
3-voisins-plus-proches

voisins-plus-proches d'une instance x sont les K instances qui sont à des distances les plus proches de x

KNN : Principe



- ▶ Si K est trop petit : sensible au bruit
- ▶ Si K est trop grand : le voisinage peut contenir des points d'autres classes

KNN : Avantages X Limites

- ▶ L'algorithme KNN est facile à comprendre et à interpréter.
- ▶ Il est utile pour les données non linéaires.
- ▶ polyvalent : utile pour la classification et la régression

- ▶ Ils ne construisent pas les modèles explicitement
- ▶ Garde en mémoire toutes les observations afin de réaliser sa prévision
- ▶ grandes tailles de jeu de données entrainera un temps beaucoup plus large
- ▶ La classification d'instances inconnues est relativement coûteuse

SVM (machine de vecteur à support)

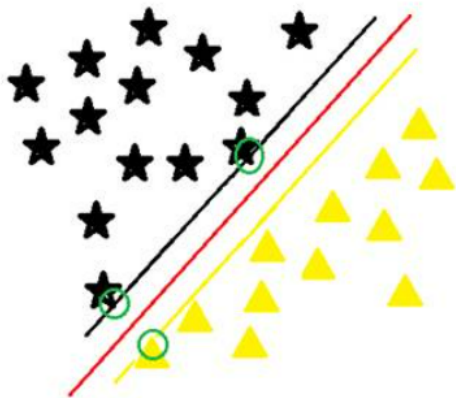
- Les SVM : séparer les données en classe à l'aide d'une frontière.
- La distance entre les différents groupes de données et la frontière séparatrice soit maximale
- Cette distance est appelée *marge*. Et les données les plus proches de la frontière sont appelées *vecteurs de support*.

Commandes :

- ▶ `from sklearn.svm import SVC`
- ▶ `from sklearn.svm import LinearSVR`

SVM

- ▶ La frontière est la droite en rouge.
- ▶ Les vecteurs de support sont les éléments vert
- ▶ La marge : la distance entre la droite rouge et les deux droites noire et jaune.



SVM : Sklearn

- ▶ Pour la classification, la bibliothèque scikit-learn met en place trois classes : SVC, NuSVC et LinearSVC.
 - ▶ NuSVC : permet de faire la classification multi-classes
 - ▶ LinearSVC : plus adapté au grand nombre d'échantillons
-
- ▶ Pour la régression, la bibliothèque scikit-learn met en place trois classes : SVR, NuSVR et LinearSVR.
 - ▶ NuSVR : la régression vectorielle de support, similaire au NuSVC et utilise le paramètre *nu* pour contrôler le nombre de vecteurs de support.
 - ▶ LinearSVR : plus adapté au grand nombre d'échantillons

SVM : Sklearn

Pour la classification :

- ▶ `from sklearn.svm import SVC`
- ▶ `model_SVC = SVC(kernel = 'linear', gamma = 'scale', shrinking = False,)`
- ▶ Les mêmes attributs du SVC sont valables pour le NuSVC.

Pour la régression :

- ▶ `from sklearn.svm import SVR, NuSVR, LinearSVR`
- ▶ `model_SVR = SVR(kernel = 'linear' , gamma = 'auto')`
- ▶ `NuSVR(kernel = 'linear', gamma = 'auto', C = 1.0, nu = 0.1)`

SVM : Applications

- ▶ Reconnaissances des formes
- ▶ Reconnaissance de chiffres manuscrits.
- ▶ Reconnaissance de visages.
- ▶ Diagnostic médical

SVM : Avantages X Limites

- ▶ Les classificateurs SVM sont d'une précision élevée et effectuent des prédictions plus rapides que d'autres modèles. Ils utilisent aussi moins de mémoire car ils utilisent un sous-ensemble de points d'entraînement dans la phase de décision.
- ▶ Les SVM ne sont pas adaptés aux grands ensembles de données puisqu'ils prennent un temps de formations plus élevé. Et ils fonctionnent mal avec les classes qui se chevauchent et dépendent du noyau utilisé.