Le machine learning

Vue d'ensemble

Rafik LACHAAL

Qu'est-ce que le machine learning

- C'est un Terminator redoutable ?
- Si je télécharge Wikipédia je fais apprendre mon ordinateur ?
- Pouvez-vous en donner une définition claire et concise ?

D'après Tom Mitchell, 1997 :

« Etant donné une tâche T et une mesure de performance P, on dit qu'un programme informatique apprend à partir d'une expérience E si les résultats obtenus sur T, mesurés par P, s'améliore par l'expérience »

Une définition plus vulgarisé :

Le « machine learning », ou **l'apprentissage automatique**, est l'art de construire des système pouvant **apprendre à partir de données**. Apprendre signifie s'améliorer sur certaine tâches, compte tenu d'une **mesure de performance**.

Un exemple : investir dans une startup

Un font d'investissement souhaite détecter quels sont les startups les plus prometteuses afin d'y investir.

Pour ce faire elle dispose d'un historique concernant 50 startup Américaines contenant :

- Les dépenses en R&D
- Les dépenses lié à l'administration de l'entreprise
- Les dépenses en marketing
- L'état ou se situe la startup (Américain)
- Le profit générer

Elle a donc développé un programme qui peut apprendre de ces données sur les 50 entreprises et donner une estimation du profit d'une nouvelle entreprise en fonction des investissement qu'elle ferait.

Un exemple : investir dans une startup

Les exemples utilisés par le système pour son apprentissage constitue le **jeu d'entraînement** (*training set* en anglais).

Chacun d'eux s'appelle une **observation d'entraînement** (on parle aussi d'échantillon).

Dans nôtre cas :

- La tâche **T** est la prédiction du profit d'une startup
- L'expérience **E** est constitué par les données d'entraînement
- La mesure de performance **P** reste à définir, mais on peut facilement s'imaginer que se serait la différence entre les valeurs prédites et les valeurs réelles.

L'intérêt du machine learning

Il est possible qu'**à une date t** les entreprises qui investissent massivement dans la recherche et le développement pour développer une nouvelle technologie soit avantagées et ai un profit plus élevées que les autres.

Quelques années plus tard, on peut atteindre un pic technologique et ne plus avoir besoin d'investir dans la R&D.

Les entreprises qui seront avantagé seront donc celle qui auront la meilleur communication et qui investissent le plus dans le marketing.

On donc le programme qui données de bonnes prédiction quelques années plus tôt qui **n'est plus aussi efficace maintenant**.

La solution est que notre programme apprennent continuellement avec de nouvelles données afin de **mettre à jour ces règles de prédictions**.

L'intérêt du machine learning

En résumé l'apprentissage automatique est excellent pour :

- Les problèmes pour les quels les solution existantes requièrent **beaucoup d'ajustements** fins ou de longue liste de règles
  - Un apprentissage automatique peut souvent simplifier le code et donner de meilleurs résultats que l'approche traditionnelles.
- Les problèmes complexes pour les quels il n'existent **aucune bonne solutions** si l'on adopte une approche traditionnelle.
- Les environnements fluctuants : un système d'apprentissage automatique peut s'adapter à **de nouvelles données**
- L'exploration des **problèmes complexes** et de **gros volumes de données**.

Les types de données

Type de données	Opérations supportées
Quantitatives continues	Calculs, égalités/différence, infériorité/supériorité
Quantitatives discrètes	Calculs, égalités/différence, infériorité/supériorité
Qualitatives nominales	égalités/différence
Qualitatives ordinales	égalités/différence, infériorité/supériorité

En apprentissage automatique **un attribut** est un type de données (ex : l'investissement en R&D) tandis qu'une **caractéristique** désigne en général un individus et ça valeur (ex : investissement R&D = 150000)

Types de système d'apprentissage automatique

On peut classer les système d'apprentissage en grande catégories :

- Selon que l'apprentissage s'effectue ou non sous supervision humaine :
  - apprentissage supervisé
  - non supervisé
  - semi-supervisé
  - avec renforcement
- Selon que l'apprentissage s'effectue ou non progressivement, au fur et mesure :
  - apprentissage en ligne
  - apprentissage groupé

Supervision humaine ou non

- **Les algorithmes supervisés :**
  - Cherchent à extraire des connaissances à partir d'un ensemble de données contenant des **couples entrées-sortie**.
  - Ces algorithmes on pour but de définir une représentation compacte des associations entrée-sortie par l'intermédiaire d'une **fonction de prédiction**.
- **Les algorithmes non supervisés** :
  - Toutes les données sont équivalente, **pas de couple entrée-sortie**.
  - Ces algorithme cherchent à **organiser des données en groups** de telle manière à ce que les données ayant des caractéristiques similaire se retrouvent dans un même groups et les données différentes dans des groups distincts.
- **Les algorithmes semi-supervisés :**
  - Les données sont partiellement étiquetées. Pour certaines observation on a un couple entrée-sortie et pour d'autre on a pas de valeur de sortie.
  - On va donc combiner des d'algorithmes supervisés et non-supervisés.
- **Apprentissage par renforcement :**
  - Le système, appelé agent dans ce contexte, peut observer son environnement et accomplir des actions.
  - En retour de chacune de ses action il obtient une réponse positive ou négative et peut alors apprendre par lui-même quel est la meilleur stratégie.

Progressivement, au fur et mesure

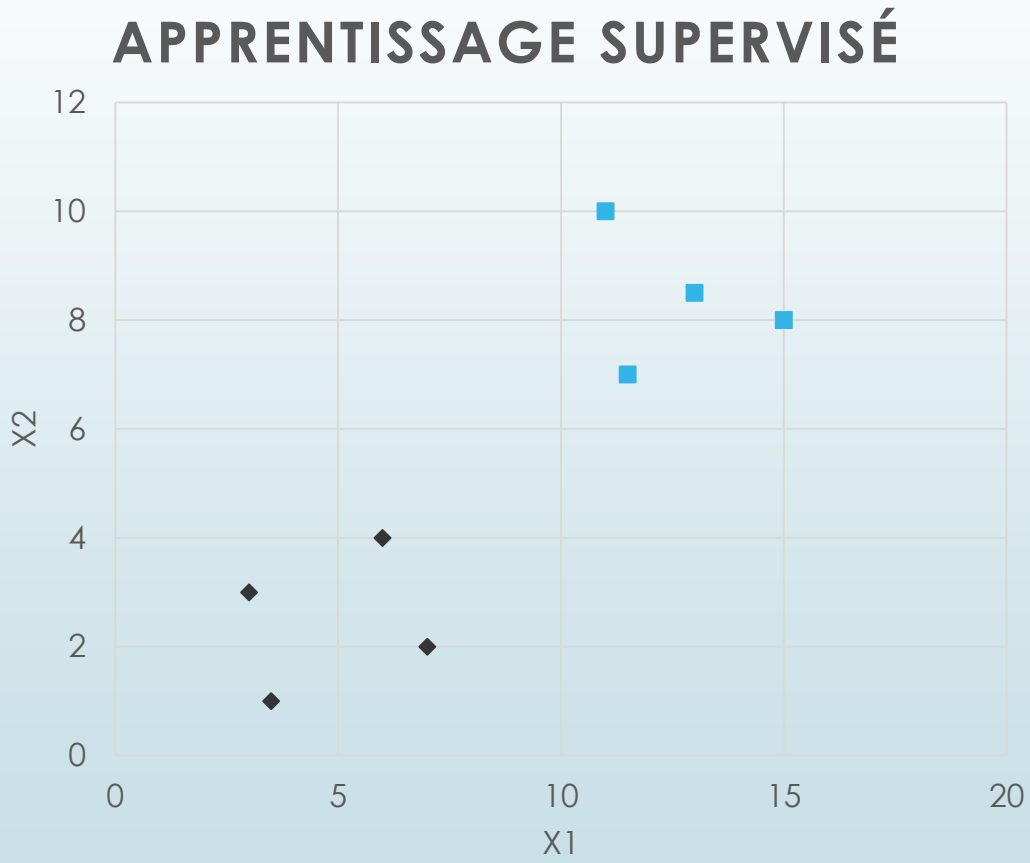
- **Apprentissage groupé (Batch learning) :**
  - Le système est incapable d'apprendre progressivement. Il doit être entraîner avec toute les données disponible.
  - Simple à mettre en place, mais nécessite beaucoup de temps et de ressource informatique.
  - Si on veut prendre en compte de nouvelles données il faut ré-entraîner le modèle avec toutes les données.
- **Apprentissage en ligne (online learning) :**
  - Le système apprend avec de nouvelles données au fur et à mesures
  - Chaque étape d'apprentissage et économique et rapide.
  - Adapter aux système de cours boursier qui doivent s'adapter et évoluer rapidement.

Les Algorithmes supervisés et non supervisés

Dans le cas des algorithmes supervisés prenons pour exemple:

- 2 variables X_1et X_2 aux quels on adjoint une **variable de sortie Y** qui pourra prendre deux valeurs {Noire, Bleu}
- Généralement on dit da la variable Y que c'est **la variable cible**, ou qu'elle est **la variables étiquettes** ou encore en anglais **le labels**.
- L'algorithme proposera donc une fonction de prédiction de la forme :

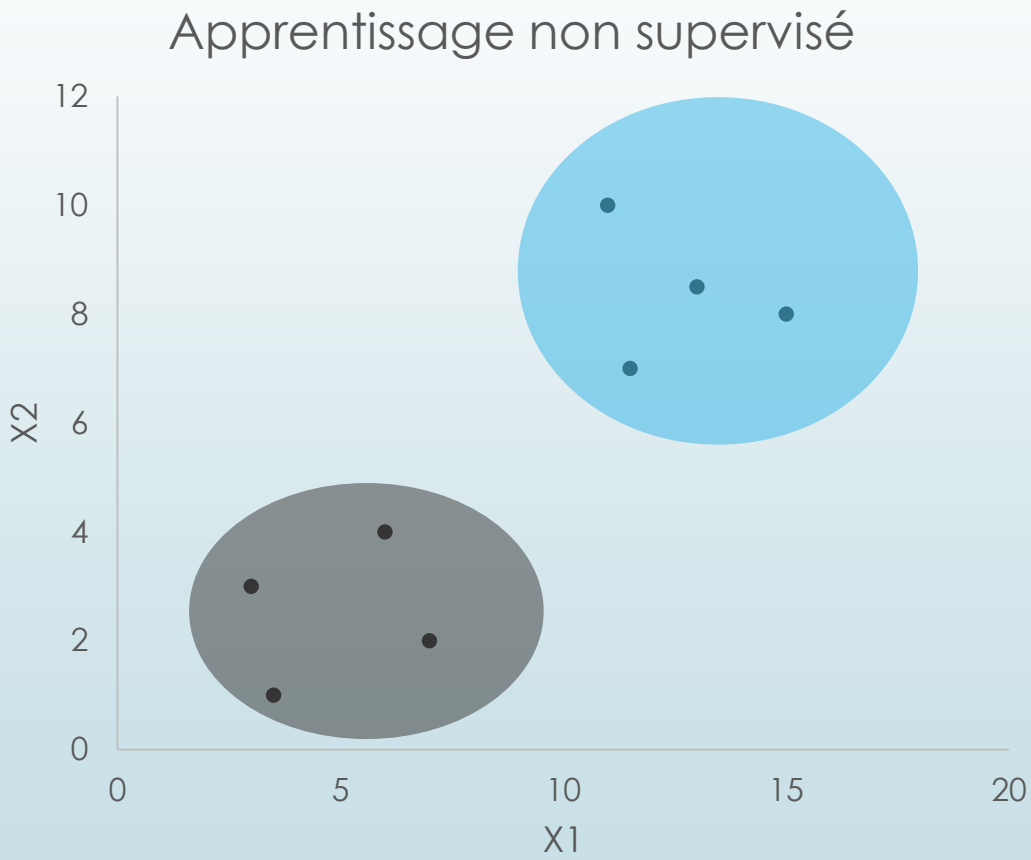
$$Y = f(X_1, X_2)$$



Les Algorithmes supervisés et non supervisés

Dans le cas des algorithmes non supervisés prenons pour exemple:

- 2 variables X_1et X_2 **sans avoir de variable de sortie Y**, donc plus de distinction de couleurs
- Les données d'apprentissage ne sont donc pas étiquetés et le système apprend sans professeur.
- L'algorithme proposera deux group (ou plus) d'individus distincts en **cherchant à regrouper les point les plus proches** dans un même group.



Les Algorithmes supervisés et non supervisés en exemple

Si on reprend l'exemple de notre jeu de donnée contenant l'historique de 50 startups :

- La variable profit serait la variables cible ou le labels
- L'investissement en R&D et l'investissement en marketing seraient les variables prédictives.

Un **algorithme supervisé** est donc le plus adapté pour traiter se problème.

Si par exemple vous disposé de nombreuse données sur les visiteurs de votre blog, vous pourriez effectuer **un partitionnement** (en anglais, *clustering*) piur tenter de détecter des groupes de visiteurs similaire.

A aucun moment vous dites à l'algorithme à quel groupe un visiteur appartient. Il peut remarquer que:

- 40% de vos visiteurs sont des hommes adorant les bandes dessinées et lisant généralement votre blog le soir.
- 20% sont des jeunes passionné de science-fiction qui le consulte durant le week-end.
- …

Le clustering est un **algorithme non-supervisé**.

Algorithme supervisés : Le choix entre la régression et la classification

On peut subdiviser les algorithmes supervisés en deux types en fonction de la valeur à prédire :

- **Les algorithmes de régression** : la valeur Y à prédire prends ses valeurs dans l'ensemble continu des réelle. Donc Y peut prendre une infinité de valeurs.

- **Les algorithmes de classification** : Y prend un nombre fini k de valeurs. Y = {1, 5, …,14 }, ou Y = {'oui', 'non'}

Le choix entre la régression et la classification

Exemple de régressions :

L'image ci contre pourrait répondre au problème suivant :

Quel est le prix d'une maison en fonction de sa taille ?

La variable de sortie étant le prix elle peut prendre une infinité de valeurs.

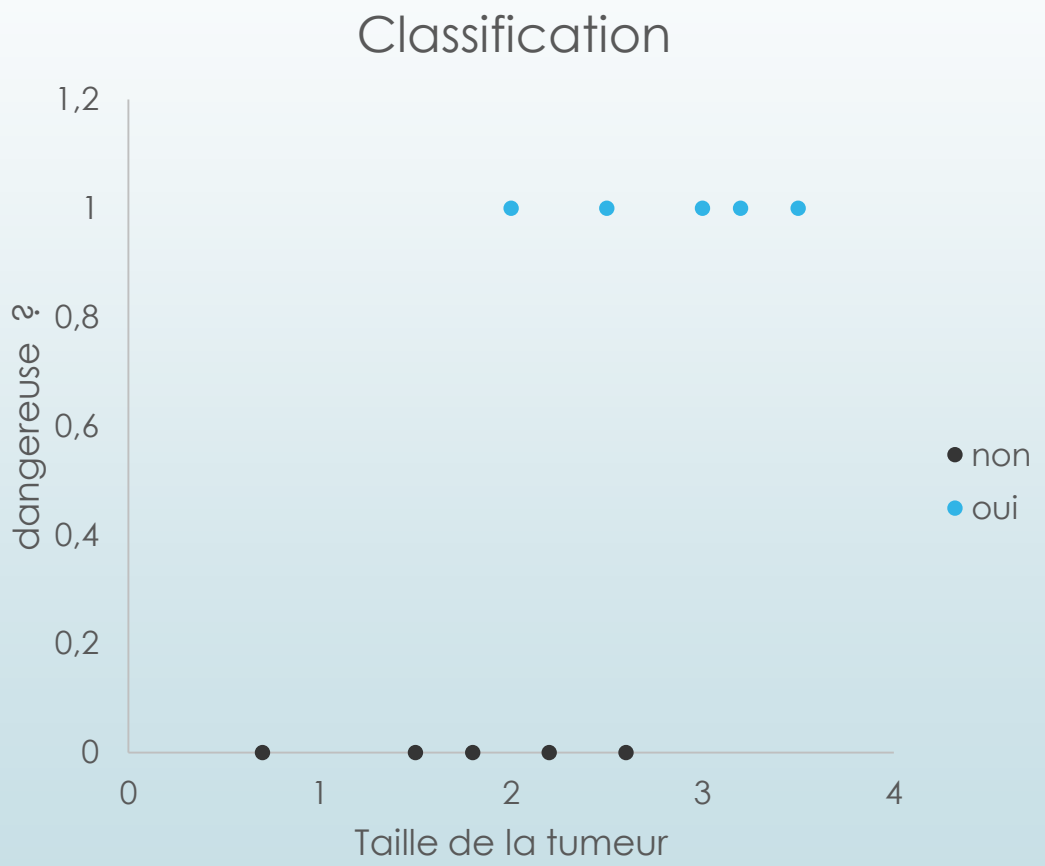
Le choix entre la régression et la classification

Exemple de classification :

L'image ci-contre pourrais répondre au problème suivant :

Une tumeur est dangereuse ou bénigne en fonction de sa taille ?

La variable de sortie ne peut prendre que 2 modalités {'oui', 'non'}



Taxinomie des algorithmes

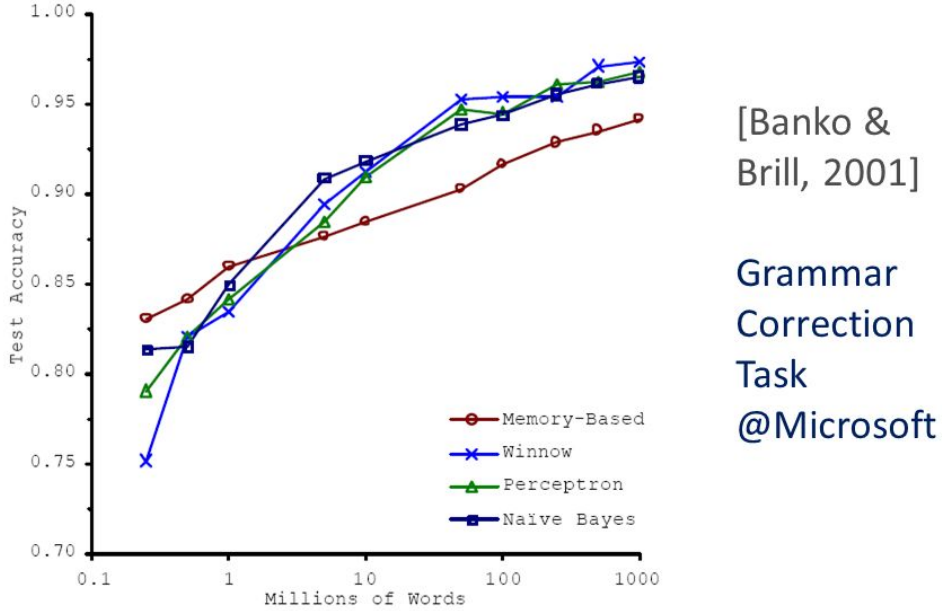
Algorithme	Mode d'apprentissage	Type de problème à traiter
Régression linéaire univariée	Supervisé	Régression
Régression linéaire bivariée	Supervisé	Régression
Régression polynomial	Supervisé	Régression
Régression régularisée	Supervisé	Régression
Naive Bayes	Supervisé	Classification
Régression logistique	Supervisé	Classification
Clustering hiérarchique	Non supervisé	-
Clustering non hiérarchique	Non supervisé	-
Analyse en composantes principales	Non supervisé	-

Principale difficulté de l'apprentissage automatique

Les données d'apprentissage sont en nombre insuffisant :

- Même pour des problème très simple il faut en générale des milliers d'exemples.

En 2001 une étude de deux chercheur de Microsoft montre que des algorithme de machine learning très différents, dont certain plutôt simples donnaient d'aussi bon résultats sur le problème complexe de la désambiguïsation sur le problème complexe du langage naturel une fois qu'ils avaient reçus suffisamment de données.



Principale difficulté de l'apprentissage automatique

Données d'entraînement non représentative :

Les données d'entraînement doivent être représentatives des future données que l'on souhaite prédire.

- Les 50 startups doivent représenter tout les secteurs d'activités dans la même proportion.
- Toutes les tailles d'entreprises.
- Tout les capitaux de départ.

Principale difficulté de l'apprentissage automatique

Le sur-apprentissage :

- Diversités de modèles applicables pour un même problème

  ⟶ Le quels choisir ?

- Les hyperparamètres : Variables contenu dans le modèle et qui permettent d'affiner son fonctionnement.
  - Pour la régression linéaire on a par exemple le nombre de variables à garder
  - Pour les KNN on le nombre de clusters que l'on désir garder.

Principale difficulté de l'apprentissage automatique

Classiquement on évalue sont modèle en appliquant le modèle prédictif sur les données qui nous ont servit pour l'apprentissage.

- Pour la régression linéaire on juge de la qualité du modèle en regardant la distance entre les prédictions et les observations empiriques

- Pour la régression logistique regarde le nombre observations bien classées.

Principale difficulté de l'apprentissage automatique

Problème, un modèle linéaire se constitue de la manière suivante :

$$Y = f(X) + \varepsilon(X)$$

- Y : Valeur de sortie que l'on cherche à prédire
- $f(X)$: Fonction de prédiction que le data analyst cherche à définir
- $\varepsilon(X)$: Fonction résiduelle aussi appelé erreur ou bruit

Le résidus ε est la partie du modèle qui ne peut être expliqué par les variables d'entrées car il est propre à chaque observations.

Principale difficulté de l'apprentissage automatique

Le risque est qu'en cherchant à valider le modèle sur les données qu'on lui à fait apprendre est qu'on lui face apprendre la fonction résiduelle au passage.

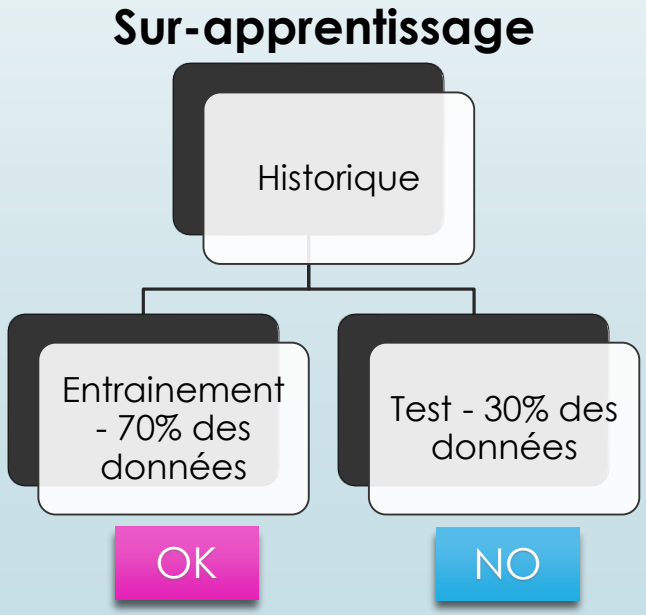
Résultat, le modèle n'est capable de faire des prédictions correctes que sur les observations que l'on lui à fait apprendre.

C'est ce que l'on appel le **sur-apprentissage.**

Principale difficulté de l'apprentissage automatique

Pour palier à ce problème -> « Testset validation »

- La première idée est de diviser son jeu de données en deux partie :
  - 70% des données utilisées pour l'apprentissage
  - 30% des données utilisées pour le test du modèle et le calcule de la performance



Principale difficulté de l'apprentissage automatique

Encore mieux : « Apprentissage, validation , test »

Diviser son jeu de données en trois partie :

- 60% des données pour l'apprentissage
- 20% pour la validation de l'ajustement du modèle (nouvel étape que l'on réitère tant que le modèle n'est pas satisfaisant)
- 20% pour le test final du modèle pour être sur qu'il n'y à pas de sur-apprentissage au moment de la validation.

Quelques sources de données

- https://www.kaggle.com/datasets
- https://registry.opendata.aws/
- https://dataportals.org/
- https://opendatamonitor.eu/
- https://www.quandl.com/
- https://www.homl.info/10
- https://www.reddit.com/r/datasets/

Questions

1. Le problème de détection de spam est-il un problème d'apprentissage supervisé ou non supervisé ?
2. Quel types d'algorithme utiliseriez-vous pour segmenter vos client en plusieurs groupes ?