Mes notes - Science des données et ML

Table des matières

A) Science des données - bases

C'est un domaine qui mélange stats, info et connaissances métier pour tirer des infos utiles des données.

4 piliers principaux:

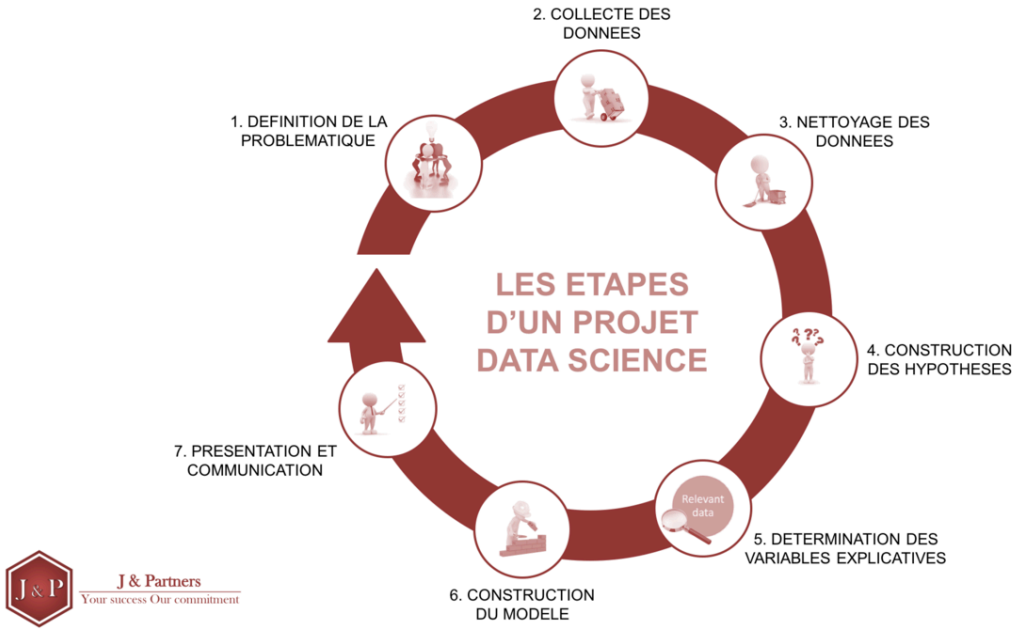
- Stats (les maths derrière tout ça)
- Informatique (outils techniques)
- Expertise métier (pour comprendre le contexte)
- Dataviz (pour rendre les résultats compréhensibles)

En gros, on suit ce processus:

1. On définit un problème
2. On collecte et nettoie les données
3. On explore et visualise
4. On construit des modèles
5. On interprète les résultats
6. On prend des décisions

Sources:

- DataScientest - Qu'est-ce que la Data Science?
- Centrale Supélec - Introduction à la Data Science



B) ML vs Deep Learning

Machine Learning

Permet aux machines d'apprendre sans être programmées explicitement. On a plusieurs approches:

- Supervisé (avec étiquettes)
- Non supervisé (sans étiquettes)
- Renforcement (apprentissage par récompense/punition)
- Semi-supervisé (mix des deux)

Deep Learning

Sous-ensemble du ML qui utilise des réseaux de neurones à plusieurs couches. Les grands types:

- CNN pour les images
- RNN pour les séquences
- GAN pour générer des choses
- Transformers pour le langage (comme GPT)

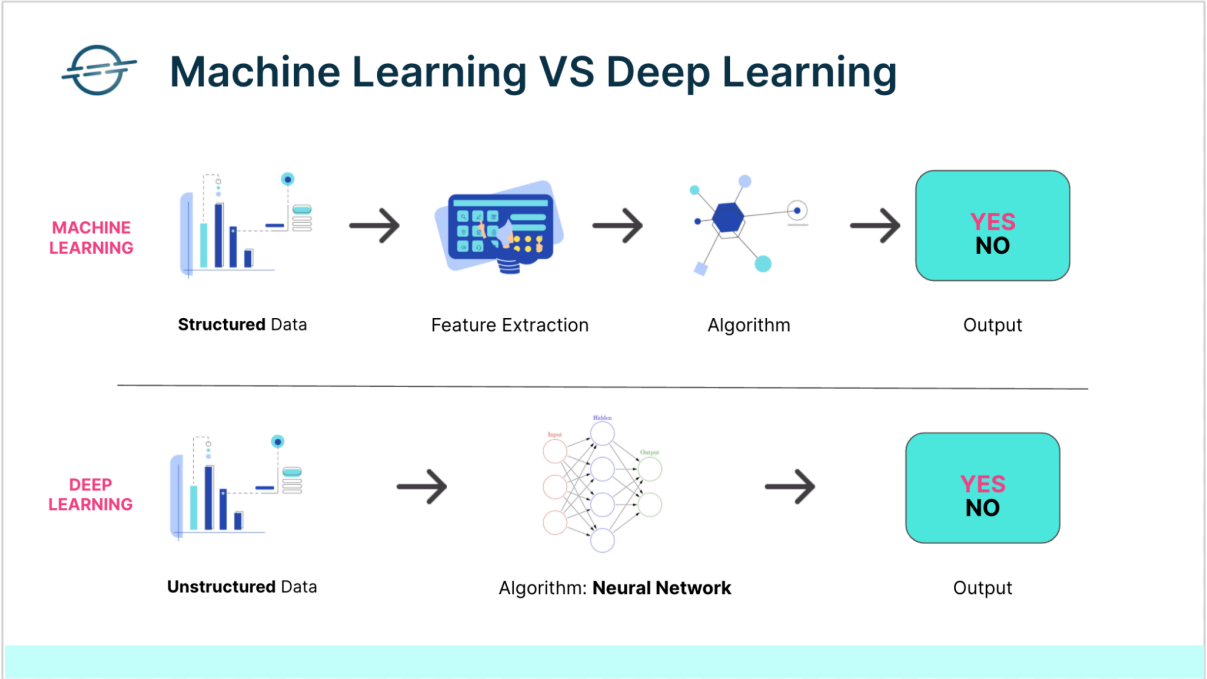
Différences importantes:

- Deep learning = besoin de BEAUCOUP de données

- Modèles deep = boîtes noires (difficiles à interpréter)
- Deep = extraction auto des features (pas besoin de le faire manuellement)
- Deep = besoin de machines puissantes (GPU/TPU)

Sources:

- INRIA - Machine Learning et Deep Learning
- OpenClassrooms - Différences entre Machine Learning et Deep Learning



C) Apprentissage supervisé

L'apprentissage avec des données étiquetées. On donne au modèle des exemples avec la bonne réponse pour qu'il apprenne les patterns.

Comment ça marche:

- On fournit des exemples X → Y
- L'algo cherche la relation entre X et Y
- Puis on peut prédire Y pour de nouveaux X

Applications:

- Classer des emails (spam/non-spam)
- Diagnostics médicaux
- Reconnaissance de visages
- Traduction

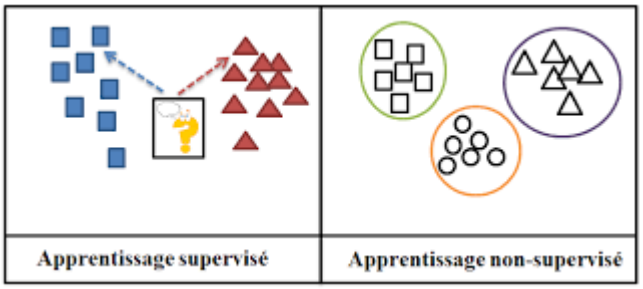
Algos populaires:

- Régression linéaire/logistique

- Arbres de décision, Random Forest
- SVM
- Réseaux de neurones

Sources:

- Université Paris-Saclay - Apprentissage supervisé
- ENSAE - Cours d'apprentissage supervisé



D) Apprentissage non supervisé

Apprentissage sans étiquettes. Le système doit trouver tout seul les structures cachées.

Objectifs:

- Trouver des groupes naturels
- Réduire la dimensionnalité
- Trouver des anomalies
- Modéliser des distributions

Applications:

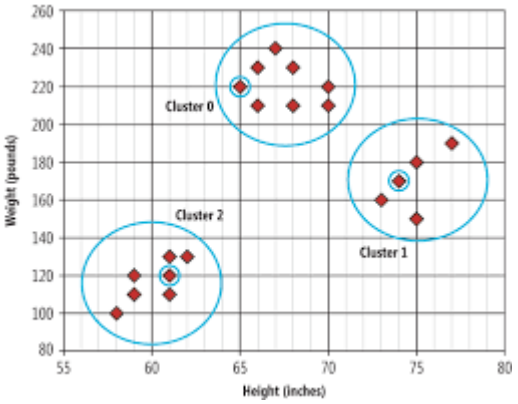
- Segmentation client
- Analyse comportementale
- Compression d'images
- Détection de fraudes

Algos:

- K-means, K-medoids
- DBSCAN, clustering hiérarchique
- PCA pour réduire les dimensions
- t-SNE, UMAP pour visualiser
- Autoencodeurs

Sources:

- Le Big Data - Guide de l'apprentissage non supervisé
- Mines ParisTech - Algorithmes de clustering



E) Classification supervisée

Un type d'apprentissage supervisé où on veut prédire une catégorie.

Caractéristiques:

- Variable cible = catégorie (pas un nombre continu)
- Output = probabilités d'appartenance
- Peut être binaire (oui/non) ou multi-classes

Métriques d'évaluation:

- Précision, rappel, F1
- ROC, AUC
- Matrice de confusion
- Accuracy

Algos courants:

- Régression logistique
- Random Forest
- SVM
- KNN
- Réseaux de neurones
- Naive Bayes

Exemple: Classer les emails en spam/non-spam

Sources:

- Télécom Paris - Évaluation des modèles de classification
- Polytechnique Montréal - Classification et métriques

	Hypothèse H_0 vraie	Hypothèse H_1 vraie
Hypothèse H_0 acceptée	Vrai positif	Faux positif
Hypothèse H_1 acceptée	Faux négatif	Vrai négatif

F) Classification non supervisée (clustering)

Le clustering regroupe des objets similaires sans connaître les étiquettes.

Objectifs:

- Regrouper des trucs similaires
- Découvrir des structures naturelles
- Simplifier les données

Évaluation:

- Indice de silhouette
- Davies-Bouldin
- Inertie (somme des carrés)
- Expertise métier

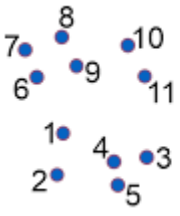
Algorithmes:

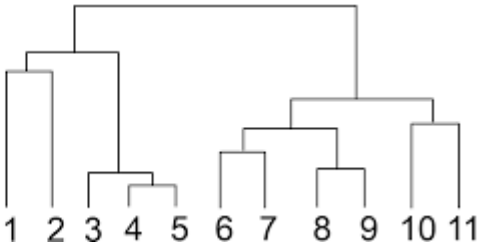
- K-means (le plus utilisé)
- Clustering hiérarchique
- DBSCAN (bon pour formes complexes)
- Mélanges gaussiens

Exemple: Segmenter des clients e-commerce selon leurs comportements d'achat

Sources:

- École Polytechnique - Algorithmes de clustering
- Sorbonne Université - Méthodes de clustering





G) Régression

Technique pour prédire une valeur numérique continue.

Types:

- Régression linéaire simple (une seule variable)
- Régression linéaire multiple (plusieurs variables)
- Régression polynomiale (relations non linéaires)
- Ridge et Lasso (avec régularisation)

Métriques:

- MSE (erreur quadratique moyenne)
- RMSE (racine de MSE)
- MAE (erreur absolue)
- R^2 (coefficient de détermination)

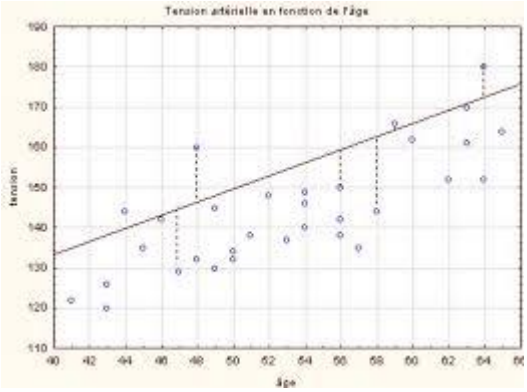
Applications:

- Prédiction de prix
- Prévision de ventes
- Analyse de tendances
- Modélisation de relations

Exemple: Prédire le prix d'une maison selon surface, quartier, etc.

Sources:

- Institut Mines-Télécom - Méthodes de régression
- Université de Lyon - Modèles de régression et évaluation



H) Validation croisée

Technique pour évaluer la performance d'un modèle sur des données indépendantes.

Types:

- k-fold (divise les données en k sous-ensembles)
- LOOCV (leave-one-out)

- Stratifiée (garde les proportions des classes)
- Temporelle (respecte la chronologie)

Avantages:

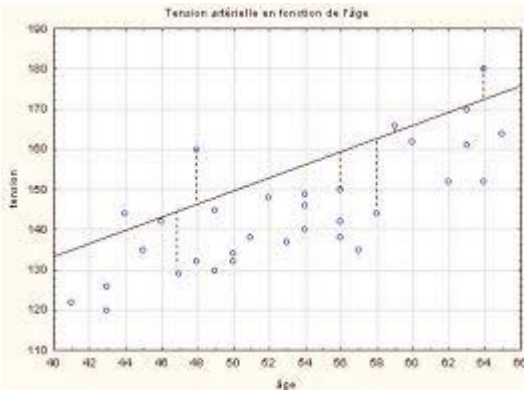
- Utilise mieux les données dispo
- Estimation plus robuste
- Détecte l'overfitting
- Permet de comparer des modèles

Comment ça marche (k-fold):

1. On divise les données en k morceaux
2. Pour chaque morceau i: a. On entraîne sur tous les morceaux sauf i b. On évalue sur le morceau i
3. On moyenne les résultats

Sources:

- ENSIMAG - Validation croisée et évaluation de modèles
- École des Ponts ParisTech - Techniques de validation



I) Données train/test/validation

Super important de bien séparer les données:

Training set (60-80%):

- Pour entraîner le modèle
- Ajuster les paramètres

Validation set (10-20%):

- Ajuster les hyperparamètres
- Surveiller les performances pendant l'entraînement
- Détecter l'overfitting

Test set (10-20%):

- Évaluation finale du modèle
- Simule des données réelles jamais vues
- NE PAS TOUCHER avant la fin!

Importance:

- Évite le data leakage (triche)
- Évaluation honnête
- Détecte over/underfitting

Méthodes de division:

- Aléatoire
- Stratifiée (préserve les proportions)
- Temporelle (pour séries chronologiques)

Sources:

- Université Paris Dauphine - Division des données
- Centrale Marseille - Best practices pour l'évaluation de modèles

J) Corrélation de Pearson

Mesure la relation linéaire entre deux variables.

Formule: r = cov(X,Y) / (σX × σY)

Interprétation:

- r entre -1 et 1
- r = 1: parfaite corrélation positive
- r = -1: parfaite corrélation négative
- r = 0: pas de corrélation
- |r| > 0.7: forte corrélation
- 0.3 < |r| < 0.7: corrélation moyenne
- |r| < 0.3: faible corrélation

Limites:

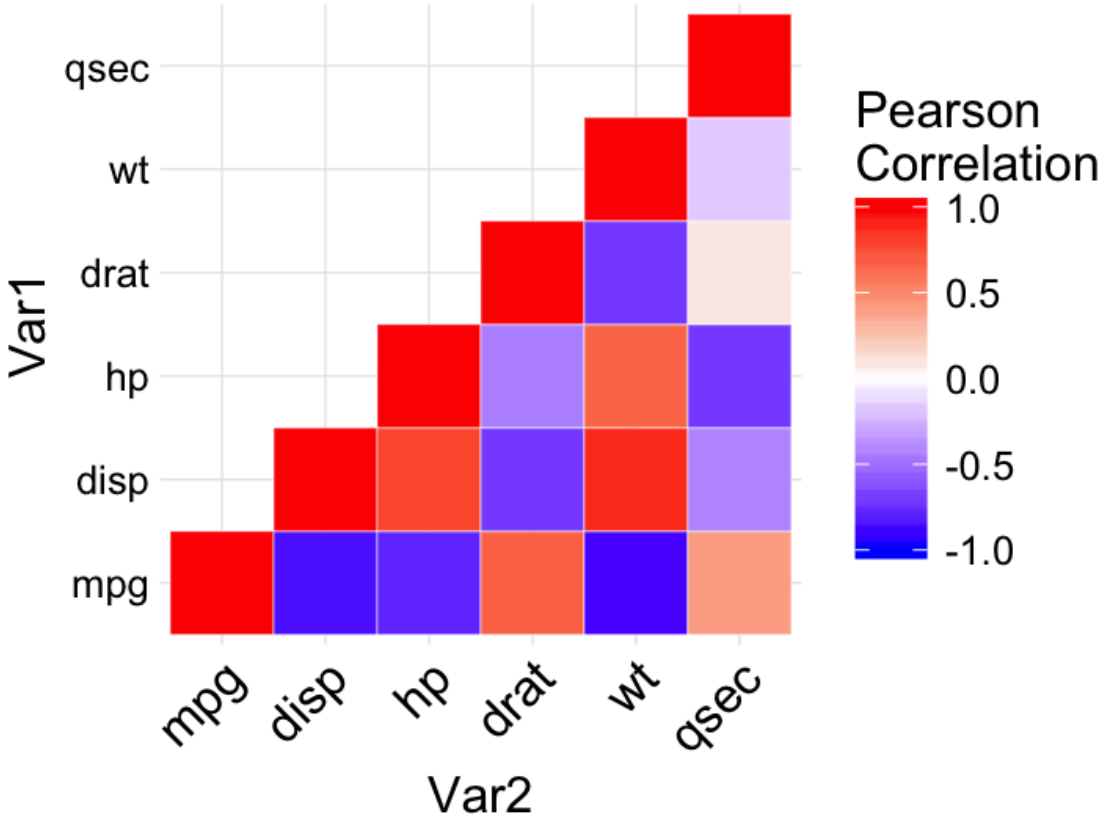
- Mesure UNIQUEMENT les relations linéaires!
- Sensible aux outliers
- Corrélation ≠ causalité!!!
- Peut manquer des relations non-linéaires

Applications ML:

- Sélection de features
- Détecter la multicolinéarité
- Réduction de dimension
- Exploration de données

Sources:

- CNAM - Statistiques et corrélations
- Université de Strasbourg - Mesures de corrélation



K) Fonctions de coût

Mesure l'écart entre prédictions et valeurs réelles. On veut la minimiser pendant l'entraînement.

Caractéristiques:

- Minimale quand prédictions = parfaites
- Convexe si possible (facilite l'optimisation)
- Dérivable (pour utiliser le gradient)
- Adaptée au problème

Fonctions courantes: Pour régression:

- MSE (moyenne des carrés des erreurs)
- MAE (moyenne des erreurs absolues)
- MSLE (pour distributions log)

Pour classification:

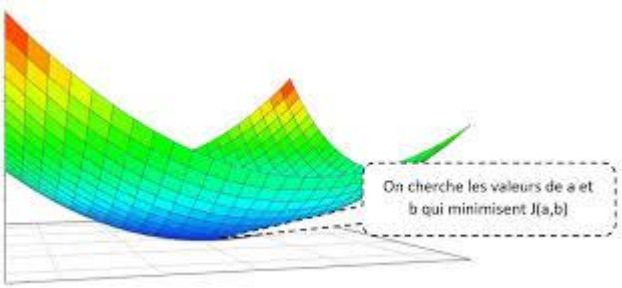
- BCE (entropie croisée binaire)
- Entropie croisée catégorielle
- Hinge loss (SVM)

Rôle dans l'entraînement:

1. Le modèle prédit
2. On mesure l'erreur avec la fonction de coût
3. On ajuste pour minimiser
4. On répète jusqu'à convergence

Sources:

- Télécom SudParis - Fonctions de coût en machine learning
- École Normale Supérieure - Optimisation et fonctions objectifs



L) Descente de gradient

Algo d'optimisation pour trouver le minimum d'une fonction de coût.

Principe:

1. On initialise les paramètres au hasard
2. On calcule le gradient (dérivée)
3. On met à jour les paramètres dans la direction opposée au gradient
4. On répète jusqu'à convergence

Formule: $\theta = \theta - \alpha \times \nabla J(\theta)$ (α = learning rate, $\nabla J(\theta)$ = gradient)

Variantes:

- Batch GD: utilise toutes les données
- SGD: un seul exemple à la fois (plus rapide, moins stable)
- Mini-batch: compromis entre les deux
- Versions avancées: Adam, RMSprop, etc.

Défis:

- Learning rate:
  - Trop grand = ça diverge
  - Trop petit = ça prend des années
- Minima locaux vs global
- Plateaux et points selles
- Learning rate adaptatif

Applications:

- Presque tous les modèles ML
- Crucial pour les réseaux de neurones

Sources:

- CentraleSupélec - Algorithmes d'optimisation
- EPFL - Descente de gradient et optimisation