MAINTENANCE PRÉDICTIVE

Plan du cours

- Introduction
- Modélisation statistique pour la maintenance
- Capteurs, analyse de données pour la maintenance
- Prédiction
- Organisation de la maintenance

Données longitudinales et analyse de survie

Rappels des cours précédents

> Cours 1 & 2 :
- Pour optimiser la maintenance, on a besoin (entre autre) d'estimer les lois de vie des équipements
- Les méthodes statistiques comme Weibull modélisent ces lois de vie
- Des méthodes spécifiques traitent les problématiques des covariates

> Cours 3 :
- Désormais, on récolte des données de plus en plus nombreuses, et tout au long de la vie des équipements
- On peut également prédire la durée de vie restante par des méthodes d'apprentissage plus courantes

> Le cours d'aujourd'hui :
- Les techniques de traitement de données sur des problématiques classiques :
  - Le nettoyage des données
  - La sélection de données
  - Le traitement des données catégorielles
- De nouvelles méthodes émergent très récemment, pour mixer les approches des cours 2 & 3

NETTOYAGE DES DONNÉES

SAFRAN

SAFRAN

Nettoyage des données : introduction

- **Les données brutes sont bruitées et corrompues**
  - > L'évident à ne pas oublier : enlever les doublons (sauf s'ils ont une raison!)
  - > Ce qu'on va voir aujourd'hui
    - Détection d'outliers
    - Données manquantes / incomplètes
    - Filtrage (très brièvement)

  - > Attention !
    - Il faut comprendre les données avant de les nettoyer
    - Il faut comprendre les méthodes utilisées pour les utiliser correctement

  - > Cette étape est cruciale **Better data > better algorithm** ou le fameux **Garbage In → Garbage Out**

Nettoyage des données : outliers

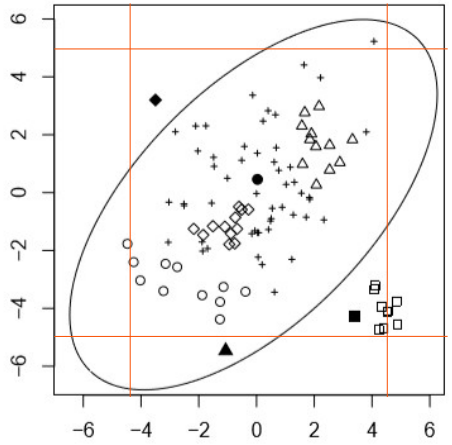
- **Détection d'outliers**

  > Outlier / anomalie → point intéressant ou bien point à retirer ?

  > L'outil classique :
  > Local Outlier Factor
  > Isolation Forest

Nettoyage des données : outliers

- **Détection d'outliers**
  - > Les outils classiques : 3 sigma | winsorizing
    - On sélectionne des bornes, tout ce qui n'est pas dans l'intervalle défini est considéré comme un outlier.
    - 3 sigma : les bornes sont à 3 écarts types de la moyenne.
      - Fonctionne bien si les données sont quasi-normales.
      - Attention, la présence d'outliers peut influencer le calcul de mu sigma …
      - On peut utiliser la médiane et le MAD (déviation médiane à la médiane) pour estimer mu sigma de façon robuste (sigma = 1.4826 · MAD pour la distribution normale).
    - Winsorizing : Os sélectionne un percentile limite. Tout ce qui dépasse est un outlier.
      - Par exemple on peut choisir 1% - 99%
    - Attention : ces outils fonctionnent sur les distributions marginales unimodales, pas sur les dépendances entre variables !
  - > Modèles multidimensionnels :
    - Local Outlier Factor
    - Isolation Forest



Nettoyage des données : outliers

- **Détection d'outliers**
  - > LOF (Local Outlier Factor)
    - Basé sur des plus proches voisins
    - La 'k-distance' : la distance qu'un point a avec son voisin k le plus éloigné
    - La distance d'atteinte (reachability distance')
      - Maximum entre 2 points et la k-distance du second point.
      - Approximation : si 2 points sont voisins, c'est la k-distance, sinon, c'est la vrai distance
    - La densité local d'atteinte (Local reachability density)

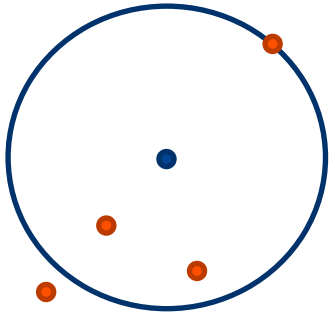
$$LRD(a) = \frac{k}{\sum(rd(a,n))}$$

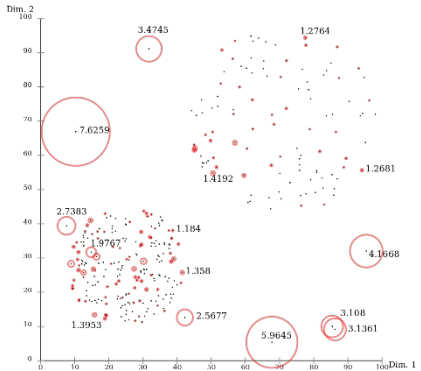
      - Intuitivement : si LRD est faible, a est loin de ses voisins, sinon, il est proche.

    - Le score LOF :

$$LOF(a) = \frac{1}{kNN(a)} \sum_{b \in kNN(a)} \frac{lrd(b)}{lrd(a)}$$

      - ~1 : A a même densité que ses voisins
      - > 1 A a une densité plus faible que ses voisins
      - < 1 A a une densité plus forte que ses voisins

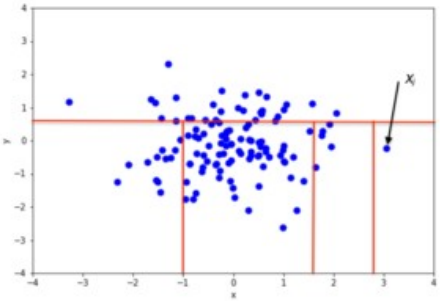


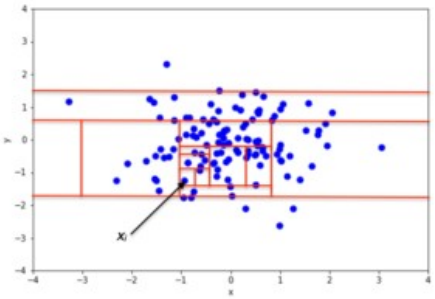


25/01/2023

Nettoyage des données : outliers

- **Détection d'outliers**
  - > Random Isolation Forest
    - Isolation Tree
      - L'ensemble des points est découpé arbitrairement sur une variable et sur une valeur de cette variable.
      - L'arbre est appris quand les points sont seuls (ou tous de la même valeur)
    - La logique ?
      - Un point isolé est plus facile à isoler
    - Score d'anomalie :
      - $s(x,\psi) = 2^{-\frac{E(h(x))}{c(\psi)}}$
      - ψ est le nombre de points dans l'ensemble d'apprentissage, voyez $c(\psi)$ comme une normalisation.
      - $h(x)$ est la longueur du chemin dans l'arbre pour isoler le point
      - Plus $h(x)$ est court, plus il a été facile d'isoler le point
      - Si ~1 -> anomalie
      - Si << 0,5 -> normal
      - Si beaucoup de sont autour de 0,5 : il n'y a peut-être pas d'outliers





Nettoyage des données : outliers

- **Quelle que soit la méthode que vous utilisez, comprenez là et comprenez ses hypothèses. Sont-elles en accord avec vos données ?**

- **On peut combiner les deux approches (univariées et multivariées)**

- **Nous avons détecté nos outliers ! Maintenant que faire ?**
  - > Les supprimer :
    - Attention à ne pas perdre trop de données.
    - Si l'outlier concerne uniquement une feature, on perd énormément d'information
  - > Remplacer par une valeur manquante :
    - Ok mais on déplace le problème
  - > Cut : on assigne à la mesure outlier la valeur max de l'intervalle admissible (3 sigma, le 95 percentile, …).
    - Simple à mettre en place sur des outliers univariés
    - Ok si peu d'outliers

- **Quelle que soit la méthode que vous utilisez, que se passe-t-il si vous rencontrez un outlier en production ? Subit-il le même traitement que lors de l'entraînement ?**

Nettoyage des données : filtrage

- **L'outil classique : la moyenne glissante**
- **D'autres opérateurs glissants :**
  - > médiane, filtrage exponentiel, …
  - > Tout type de filtres vu en cours de traitement du signal
- **Filtrages spécifiques pour données spécifiques :**
  - > Textes
  - > Images infrarouges
- **Et le filtrage par conditionnement bayésien :**
  - > Si vous avez des modèles d'évolution
  - > Kalman / filtrage particulaire

Nettoyage des données : traitement des données manquantes

- **Les trois types de valeur manquantes :**
  - > **missing** completely at random (MCAR) : le fait que la mesure soit manquante est indépendante de l'individu observé. Dans ce cas, il est impossible de prédire la valeur manquante à partir des données :
    - Exemple 1 : Un capteur n'a pas émis de signal pour une faute inconnue
  - > **missing** at random (MAR) : le fait que la mesure est manquante est dépendante du reste des données, mais pas de la valeur sous-jacente
    - Exemple 2 : Un capteur de pression renvoie NaN car il surchauffe.
  - > **missing** not at random (MNAR) : le fait que la mesure est manquante dépend de la valeur sous jacente :
    - Exemple 3 : Un capteur avionique renvoie Nan car il mesure une valeur en dehors de sa plage de réponse.

- **Il est critique d'identifier les cas probables de MAR / MNAR car leur traitement est différent. Une connaissance métier peut permettre d'identifier les sources de valeurs manquantes.**

- **Outils de diagnostiques :**
  - > Corrélation entre présence de nan et la valeur des autres colonnes
  - > Corrélation entre présence de nan et la présence des autres colonnes
  - > Paquet python missingno : https://github.com/ResidentMario/missingno

Nettoyage des données : traitement des données manquantes

- **L'outil classique : drop**
  - > Dans la réalité, si on a des données complexes : dropnan peux amener à des pertes de 80% des données !
  - > Pourquoi ?
    - Car par défaut, beaucoup d'outils de gestion de base de données sont sous forme de table
    - Les tables ne sont pas adaptées au conditionnel (ex: poumons vs maladies respiratoires).

Nettoyage des données : traitement des données manquantes

- **L'outil classique : drop**
  - > Dans la réalité, si on a des données complexes : dropnan peux amener à des pertes de 80% des données !
  - > dropnan permet d'avoir un jeu de donné 'plein' pour l'entrainement et le test, mais que ce passe-t-il si l'on rencontre un NaN en production ?
- **Indicateur de données manquantes**
  - > À utiliser en combinaison avec une autre méthode ou en supprimant la colonne contenant les Nans
  - > Créé une colonne booléenne qui vaut 1 si la donné a été remplie car manquante
  - > Adapté au traitement des MNAR
- **Complétion de données**
  - > Fillna(0)
    - si vos données sont normalisées / standardisés, équivalent a imputer la valeur moyenne.
    - Simple mais assez efficace : on rempli avec la valeur non nulle la moins informative possible.
    - A éviter si MNAR suspecté.
  - > Remplacer par une valeur tiré au hasard (par individu) dans la colonne à remplir.
    - Cela permet de préserver la distribution marginale
    - Mais pas la structure de dépendance entre variables
    - mais peut créer du signal à l' échelle d'un individu.
    - A combiner obligatoirement avec un indicateur de données manquantes.
  - > Utiliser les autres features pour prédire la feature manquante
    - Très couteux en calcul, inefficace si MNAR

SÉLECTION DE VARIABLES

SAFRAN

SAFRAN

Sélection de variables : introduction

- **Un océan de données**
  - > Car on n'a pas tout jeté
  - > Car on a bien nettoyé ces données

- **Comment choisir des données utiles pour notre problème ?**
  - > Méthodes par filtrage
  - > Méthodes « Wrapper »
  - > Méthodes « embedding »

  - > On ne parlera pas ici du cas non supervisé. Il peut être utile pour de la dataviz et/ou en recherche de cause.

Sélection de variables : méthodes par filtrage

Le principe général : classer les variables selon un certain critère, puis un seuil vient trier

En supervisé :

> Une variable n'est pas pertinente si elle est indépendante des labels
> Comment estime-t-on cette indépendance ?

Deux exemples :

> Les corrélations (ici Pearson) $R(i) = \frac{cov(x_i, y)}{\sqrt{((var(x_i)var(y))}}$

Toujours se rappeler de l'hypothèse du modèle (quelle est l'hypothèse ici ?)

> L'information mutuelle

- Pour rappel, l'entropie de Shannon : $H(Y) = -\sum p(y)\log(p(y))$
- L'entropie conditionnelle : $H(Y|X) = -\sum_X \sum_Y p(x,y)\log(p(y|x))$ observer x fait baisser l'incertitude sur y, ainsi:
- $I(Y,X) = H(Y) - H(Y|X)$
- La formule a l'air simple mais quand on n'a pas p il faut l'estimer

Sélection de variables : méthodes par filtrage

- **Voyez-vous un inconvénient avec ces méthodes proposées ?**

Sélection de variables : méthodes par filtrage

- **Voyez-vous un inconvénient avec ces méthodes proposées ?**

- **On ne filtre pas les variables qui sont corrélées entre elles !**
  > On a un conflit entre redondance vs pertinence
- **On regarde les variables une à une**
  > Certaines variables peu informatives seules mais importantes en combinaison avec une autre

Sélection de variables : méthodes par filtrage

- **Méthodes par filtrage**
  - > Illustrations

  - > Blanc / Noir -> les 2 classes
  - > Pairplot

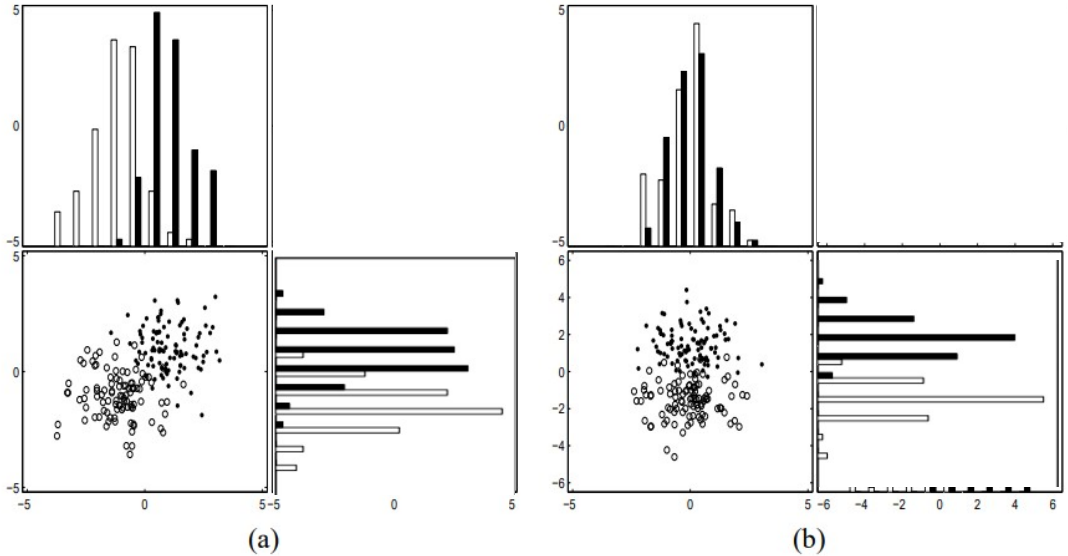


Figure 1: **Information gain from presumably redundant variables.** (a) A two class problem with independently and identically distributed (i.i.d.) variables. Each class has a Gaussian distribution with no covariance. (b) The same example after a 45 degree rotation showing that a combination of the two variables yields a separation improvement by a factor $\sqrt{2}$. I.i.d. variables are not truly redundant.

Sélection de variables : méthodes par filtrage

- **Méthodes par filtrage**
  - > Illustrations

  - > Blanc / Noir -> les 2 classes
  - > Pairplot

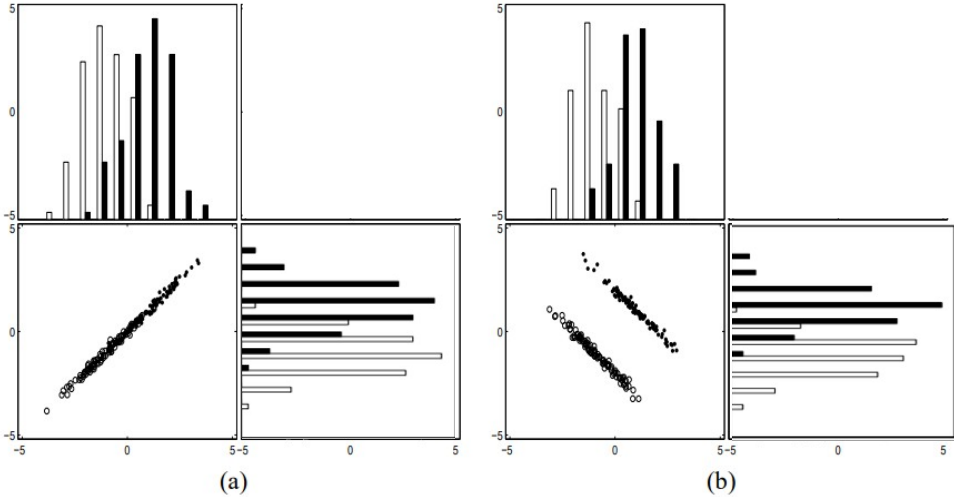


Figure 2: **Intra-class covariance.** In projection on the axes, the distributions of the two variables are the same as in the previous example. (a) The class conditional distributions have a high covariance in the direction of the line of the two class centers. There is no significant gain in separation by using two variables instead of just one. (b) The class conditional distributions have a high covariance in the direction perpendicular to the line of the two class centers. An important separation gain is obtained by using two variables instead of one.

Sélection de variables : méthodes par filtrage

- **Méthodes par filtrage**
  - > Illustrations

  - > Blanc / Noir -> les 2 classes
  - > Pairplot

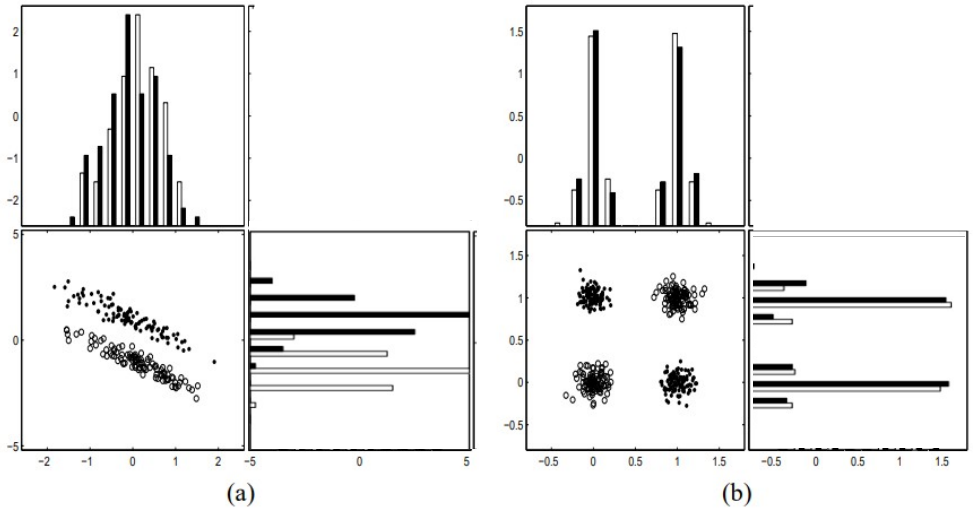


Figure 3: **A variable useless by itself can be useful together with others.** (a) One variable has completely overlapping class conditional densities. Still, using it jointly with the other variable improves class separability compared to using the other variable alone. (b) XOR-like or chessboard-like problems. The classes consist of disjoint clumps such that in projection on the axes the class conditional densities overlap perfectly. Therefore, individual variables have no separation power. Still, taken together, the variables provide good class separability .

Sélection de variables : méthodes « Wrapper »

- **Méthodes « wrapper »**
  - > Le principe : utiliser un modèle prédictif et utiliser sa performance pour évaluer les variables.
  - > Le problème principal : évaluer tous les sous-ensemble est trop coûteux, il faut donc une stratégie pour parcourir l'espace de ces sous-ensemble

  - > Deux stratégies :
    - Séquentielles
    - Heuristiques

Sélection de variables : méthodes « Wrapper »

- **Méthodes « wrapper » : séquentiel**
  - > Les algorithmes les plus simples
    - Partir d'aucune variable et ajouter des variables petit à petit (Sequential Feature Selection)
    - Partir de toutes les variables et les retirer une par une (Sequential backward Selection)
  - > Des algorithmes combinés:
    - Sequential Floating Forward Selection (à droite)
    - Adaptive Sequantial Forward Floating Selection
    - Etc, etc…

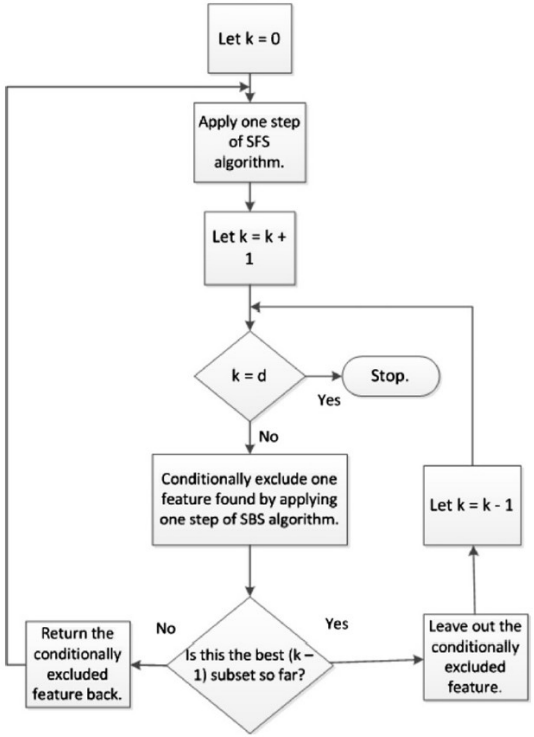


Fig. 1. SFFS flow chart.

Sélection de variables : méthodes « Wrapper »

- **Méthodes « wrapper » : heuristique**
  - > Algorithmes génétiques
    - Un individu = présence / absence d'une variable
  - > Différents acronymes pour différentes méthodes
    - CHCGA
    - PSO (Particule Swarm Optimization)
    - Etc…

  - > Quels sont les inconvénients ?
    - Beaucoup d'évaluation de modèles
    - Risque de surapprentissage

Sélection de variables : méthodes « embedding »

- **Méthodes « embedding »**
  - > Incorporer la sélection directement dans l'entrainement

  - > Exemples :
    - Reprendre l'Information Mutuelle vu quelques slides précédemment et en faire un objectif (
    - Enlever les variables dont le poids est trop faible () →
    - Ajouter dans le problème d'optimisation d'un SVM de la parcimonie (norme L1)
    - Network pruning
    - et LASSO bien sûr

VARIABLES CATEGORIELLES

SAFRAN

SAFRAN

Variables catégorielles : Introduction

- **Toutes les données ne sont pas numériques**
  - > Identifiants (code IATA aéroport)
  - > Langage naturel (texte dans un champs libre ou bien appréciation d'un opérateur)
  - > Catégories (« high », « low »)

- **Comment les numériser ?**
  - > Dépendamment des caractéristiques de la variable en considération :
    - Cardinalité
    - Forme de la distribution de fréquence (une ou quelques catégorie regroupe la majorité des observations)
    - Confidentialité

Le traitement des variables catégorielles est un travail complexe, nécessitant une connaissance forte du métier et ayant un impact important sur la performance du modèle.

Variables catégorielles : différentes méthodes

- **Ordinal encoding**
  - > Quoi : On assigne à chaque catégorie un numéro
  - > Pourquoi : On a très peu de cardinalité ou les catégories sont ordonnées (ex : « High » / « Medium » / « low »)
  - > Attention :
    - Si l'ordre n'est pas présent dans les données, on peut créer des corrélations fausses
    - Quelle rang donner à une nouvelle catégorie après entrainement ?

- **One hot encoding**
  - > Quoi : On créé une feature booléenne par catégorie (« is_A », « is_B », …)
  - > Pourquoi : On souhaite traiter indépendamment chaque catégorie
  - > Attention :
    - création potentielle de corrélations fausses
    - Explosion du nombre de feature qui peut être problématique
    - Laisser [0, 0, …, 0] pour représenter les nouvelles catégories après entrainement

Variables catégorielles : différentes méthodes

- **Hash encoding**
  - > Quoi : On utilise un algorithme de Hachage pour grouper les catégories en buckets aléatoires numérotés.
  - > Pourquoi : On cherche à réduire la cardinalité de la feature et à se protéger en cas de nouvelles catégories après entrainement
  - > Attention :
    - Hyper paramètre (nombre de bucket) à optimiser comme le reste
    - Groupe des catégories pas nécessairement équivalentes dans le même bucket

- **Embedding layer**
  - > Quoi : On utilise un algorithme pour trouver une représentation optimale de la feature catégorielle. Par exemple :
    - Autoencodeurs
    - Label encoding
    - Mean encoding
    - CatBoost
  - > Pourquoi : On cherche à modéliser la relation des catégories avec le reste des données
  - > Attention :
    - Hyper paramètres à optimiser comme le reste
    - Sujet au ‘target leaks’ donc au surapprentissage
    - Certaines méthodes gèrent mal les nouvelles catégories après entrainement

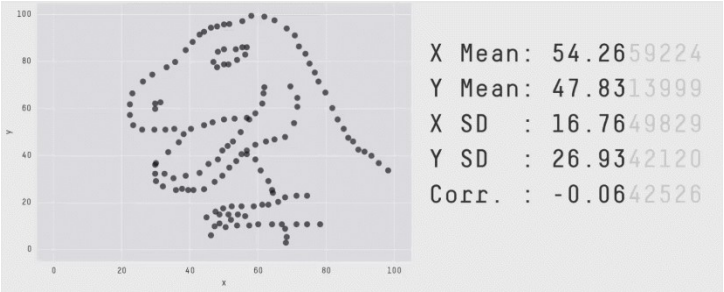
Variables catégorielles : choix de la méthode

- **La connaissance métier est l' élément le plus important**
- **Pistes de réflexion :**

Caractéristiques / méthodes	Cardinalité (nombre de valeur uniques)				Confidentialité	distribution des fréquence		Nouvelles categories
	Tres haute (> 100)	Haute	Basse (<10)	Tres basse (<5)		uniforme	très inégale (skewed)	
ordinal encoding								
one hot encoding								
hash encoding								
embedding layer								Dépends de l'embedding

- **On peut combiner plusieurs méthodes pour une même feature**
  > Attention à bien opérer une feature sélection pour éviter de surreprésenter les catégorielles

Conclusions sur les prétraitements



- **Nettoyer les données**
  - > Enlever les outliers
  - > Filtrer vos données
  - > Gérer les NaN
- **Sélectionner les variables**
  - > En amont, par filtrage ou « wrapper »
  - > Dans la méthode de régression elle-même
- **Les données catégorielles**
  - > Différentes méthodes en fonction de la nature de ces données
  - > Bien comprendre les étapes de pré-traitement pour une meilleure analyse
- **Dans tous les cas**
  - > Faites attention à toutes ces étapes, qui peuvent être clef dans la réussite ou l'échec de votre analyse
  - > Faites attention à la cohérence de ces étapes dans les différentes analyses que vous pouvez être amenés à faire

Source de l'image : The Functional Art: An Introduction to Information Graphics and Visualization: Download the Datasaurus: Never trust summary statistics alone; always visualize your data

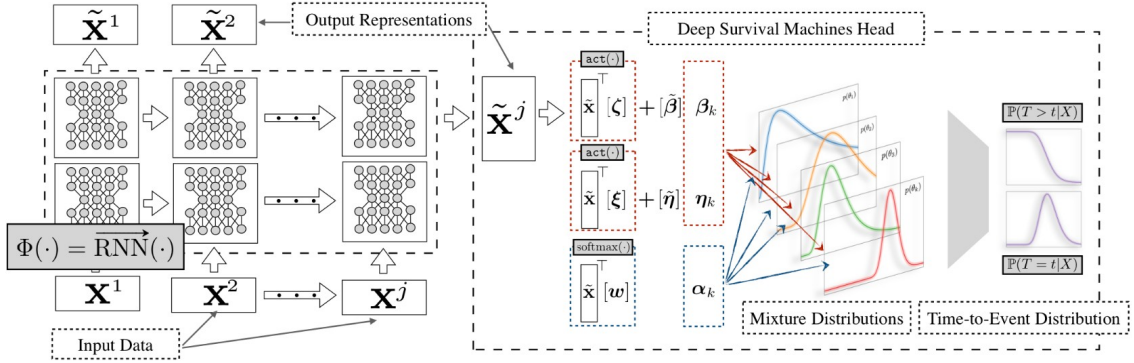
DONNÉES LONGITUDINALES ET ANALYSE DE SURVIE

SAFRAN

SAFRAN

Données longitudinales et analyse de survie

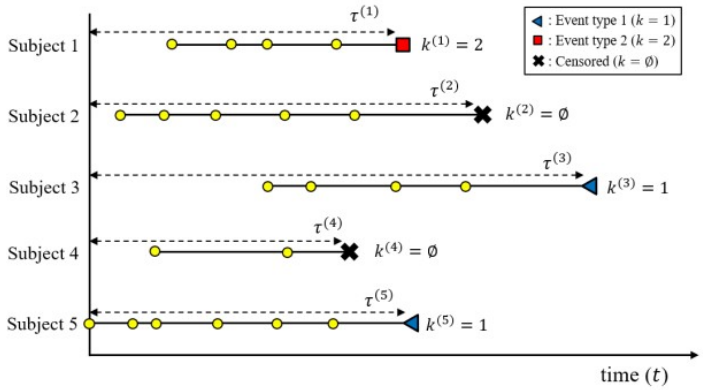
- Deep Survival Machines
  - > Une hybridation simple :
    - Un réseau de neurones pour encoder l'information
    - Un décodeur pour transformer en paramètres
    - Utiliser ces paramètres dans les distributions classiques



> Source : NAGPAL, Chirag, LI, Xinyu, et DUBRAWSKI, Artur. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*, 2021, vol. 25, no 8, p. 3163-3175.

Données longitudinales et analyse de survie

- **DeepHit / DynamicDeepHit**
  - > L'un des premiers articles s'attaquant au sujet
  - > Une notion dont on a peu parlé :
    - Les risques concurrents
  - > Les événements qui surviennent peuvent être de différentes natures
  - > Des méthodes spécifiques doivent être mises en place

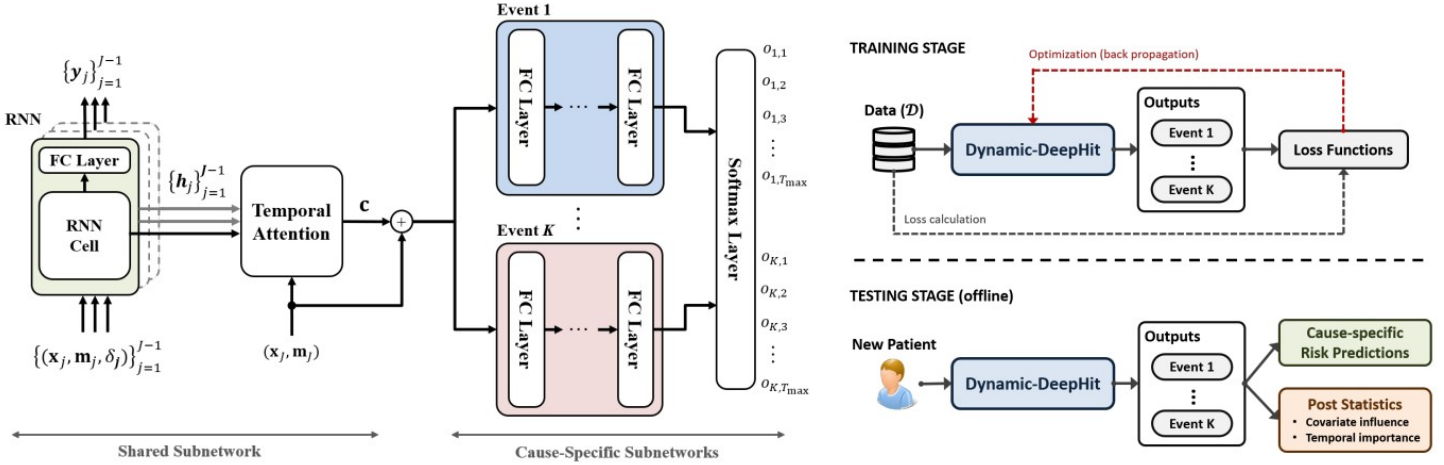


> Source : LEE, Changhee, YOON, Jinsung, et VAN DER SCHAAR, Mihaela. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 2019, vol. 67, no 1, p. 122-133.

Données longitudinales et analyse de survie

- **DeepHit / DynamicDeepHit**

  > Idée : construire un réseau par cause

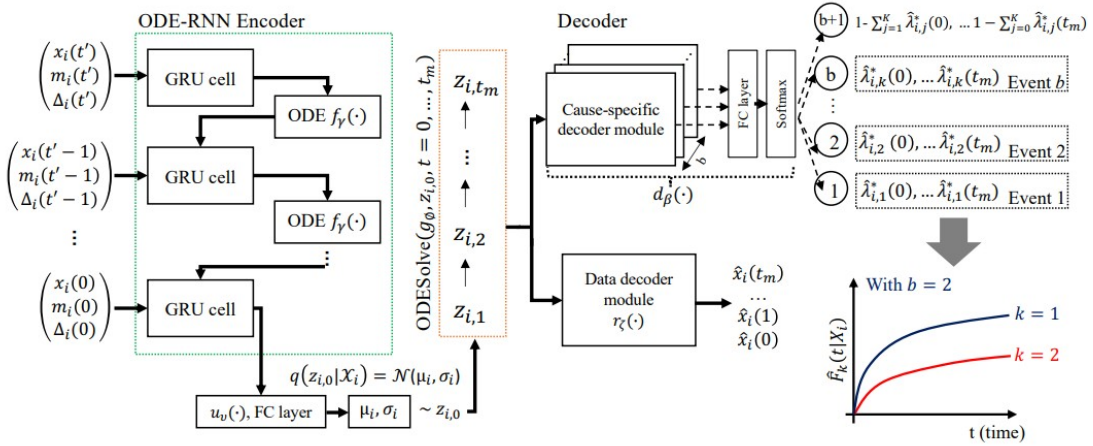


(a) The network architecture with K competing risks. (b) A schematic depiction

> Source : LEE, Changhee, YOON, Jinsung, et VAN DER SCHAAR, Mihaela. Dynamic-deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 2019, vol. 67, no 1, p. 122-133.

Données longitudinales et analyse de survie

- **SurvLatent ODE**
  - > Une autre proposition : utiliser les Neural ODE (équations aux dérivées partielles) dans l'encoder
  - > Cette méthode d'encoding permet de traiter les données manquantes nativement



> MOON, Intae, GROHA, Stefan, et GUSEV, Alexander. SurvLatent ODE: A Neural ODE based time-to-event model with competing risks for longitudinal data improves cancer-associated Deep Vein Thrombosis (DVT) prediction. *arXiv preprint arXiv:2204.09633*, 2022.

Conclusions sur données longitudinales et analyse de survie

- **Et d'autres encore**
  - > SurvSeq2Seq
  - > TSNN / RSNN
  - > …

- **Ce champ de recherche est très actif, et de nouveaux articles sortent régulièrement**
  - > La compréhension des avantages et inconvénients de ces nouvelles méthodes est encore à construire

CONCLUSIONS