Comment gérer les données manquantes ?

ML pour la santé

Oulad Ali Ayoub Mabrouk Mohy

1er avril 2026

Plan

1. **Introduction** : contexte clinique et enjeux des données manquantes.
2. **Mécanismes et structures** : MCAR, MAR, MNAR et répartitions des manquants.
3. **Imputation classique** : suppression, moyenne/médiane, KNN, NIPALS, LOESS, SVD, MICE.
4. **Imputation générative** : GAIN, VAE, VAEAC.
5. **Évaluation et impact** : qualité d'imputation, conséquences sur explicabilité et Shapley values.
6. **Guide pratique** : arbre décisionnel et recommandations.

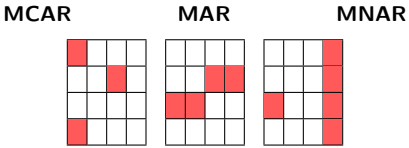
Contexte et problématique

- Les données manquantes sont un enjeu méthodologique majeur et une source potentielle de biais en recherche clinique.
- Le problème est particulièrement marqué dans les études de vie réelle, souvent touchées par des pertes de suivi et des informations incomplètes.
- L'imputation constitue un défi central, car elle influence directement la validité et la fiabilité des analyses.
- Cette revue systématique dresse un état des lieux des méthodes d'imputation basées sur le Machine Learning en santé et évalue leurs performances.

Trois mécanismes de données manquantes

- **MCAR** : *Missing Completely At Random* – absence indépendante de toute variable.
- **MAR** : *Missing At Random* – absence dépend des variables observées seulement.
- **MNAR** : *Missing Not At Random* – absence dépend des valeurs manquantes elles-mêmes.

Schéma comparatif – MCAR vs MAR vs MNAR



Cases rouges = valeurs manquantes

Probabilité d'absence (formulation)

- **MCAR :** $p(M \mid Y) = p(M)$ – les manquants sont statistiquement indépendants.
- **MAR :** $p(M \mid Y) = p(M \mid Y_{obs})$ – dépendance seulement des valeurs observées.
- **MNAR :** $p(M \mid Y)$ dépend de Y_{mis} (et pas seulement de Y_{obs}) – cas intraitable sans hypothèses fortes.

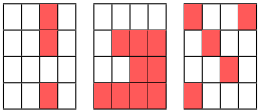
Répartition des données manquantes

Soit $Y = (y_{ij}) \in \mathbb{R}^{n \times p}$ la matrice des données et $M = (m_{ij})$ la matrice d'indication des valeurs manquantes.

- **Univariée :** les valeurs manquantes concernent une seule variable Y_k.
- **Monotone :** si Y_j est manquante pour un individu, alors toutes les variables suivantes $\{Y_k\}_{k>j}$ le sont aussi.
- **Non monotone (arbitraire) :** les valeurs manquantes n'ont pas de structure d'ordre particulière dans M.

Illustration des 3 répartitions

Univariée **Monotone** **Non monotone**



Cases rouges = valeurs manquantes

MNAR – Quand les manquants ne sont pas aléatoires

Exemple clinique : patients ayant des résultats très mauvais refusent le suivi $\Rightarrow$ les manquants dépendent de la valeur non observée.

Stratégies possibles :

- **Analyse de sensibilité :** tester différentes hypothèses sur Y_{mis} et mécanisme.
- **Modèles de sélection :** modéliser explicitement $p(M|Y)$ via copula ou pattern-mixture models.
- **Variables instrumentales :** si données externes disponibles pour identifier le biais.
- **Reconnaissance et documentation :** reconnaître MNAR et ne pas prétendre une solution universelle.

MNAR n'a pas de solution universelle sans hypothèses externes fortes.

Approches courantes – panorama

- **Suppression :** éliminer observations ou variables avec manquants.
- **Imputation simple :** moyenne, médiane, mode, last observation carried forward.
- **Imputation basée distances :** KNN (k-nearest neighbors).
- **Imputation par réduction dimensionnelle :** NIPALS, SVD, PCA.
- **Imputation itérative :** MICE (Multivariate Imputation by Chained Equations), LOESS.
- **Imputation générative :** GAIN, VAE, VAEAC (neural networks).

Retour sur la suppression de données

- **Cas complets (listwise deletion) :** éliminer individus avec ≥ 1 valeur manquante (ex. `na.omit` sous R).
  - – Perte d'information, baisse de précision.
  - – Biais si données **non-MCAR**.
- **Cas disponibles (pairwise deletion) :** utiliser observations disponibles par paire de variables.
  - \+ Préserve plus d'information.
  - – Résultats issus de sous-échantillons différents, incohérents.
- **Règle empirique :** suppression acceptable si $|M| < 5\%$ ET données MCAR (Little & Rubin, 2002).

Complétions stationnaires – valeurs "fixes"

- **CMCF** (*Most Common Attribute Value Fitting*) : imputer par la valeur la plus fréquente.
- **LOCF** (*Last Observation Carried Forward*) : pour séries temporelles.

$$(y_{ij})_{mis} = y_{ij'} \quad \text{avec} \quad m_{ij'} = 0,\ j' < j$$

- **Limites :** très simpliste, biais dans dépendances.

Imputation par combinaison linéaire

- **Moyenne :** $(y_{ij})_{mis} = \bar{Y}_j$ (moyenne marginal de la colonne j).
- **Médiane :** $(y_{ij})_{mis} = \tilde{Y}_j$ (plus robuste aux outliers).
- **Généralisation :** combinaison linéaire pondérée des observations disponibles.
- **Variantes :** agrégation locale, régression simple, hybrides.

Avantage : très rapide. **Limite :** ignore la structure de dépendance, sous-estime la variance.

Méthode des plus proches voisins (KNN)

Algorithme des k plus proches voisins (k-NN)

1. Choisir un entier k, avec $1 \leq k \leq n$.
2. Calculer les distances $d(Y_i, Y_{i'})$ (Euclidienne, Manhattan, etc.) pour $i = 1, \ldots, n$.
3. Retenir les k observations $Y_{(i_1)}, \ldots, Y_{(i_k)}$ de plus petite distance.
4. Imputer par la moyenne des k voisins :

$$(y_{ij})_{mis} = \frac{1}{k} \sum_{r=1}^{k} Y_{(i_r)}$$

Avantage : capture la structure locale. **Limite :** sensible au choix de k, calcul en $O(n^2)$.

NIPALS (Nonlinear Iterative Partial Least Squares)

Algorithme NIPALS pour réduction dimensionnelle avec imputation

1. Initialiser $Y_0 = Y$.
2. Pour $h = 1, \ldots, q$: fixer $\xi_h = Y_{h-1}^1$ (première colonne).
3. Itérer jusqu'à convergence de u_h : mise à jour de u_h, normalisation, puis mise à jour de ξ_h.
4. Mettre à jour la matrice résiduelle : $Y_h = Y_{h-1} - \xi_h u_h$.
5. Imputer les valeurs manquantes par :

$$(\hat{y}_{ij})_{mis} = \sum_{h=1}^{q} \hat{\xi}_h(i)\hat{u}_h(j)$$

Contexte : utilisé en PLS (Partial Least Squares) et analyse dimensionnelle.

Régression locale (LOESS)

Algorithme LOESS (*Locally Estimated Scatterplot Smoothing*)

1. Obtenir les k plus proches voisins $Y_{(i_1)}, \ldots, Y_{(i_k)}$.
2. Construire les matrices $A \in \mathbb{R}^{k \times (n-q)}$, $B \in \mathbb{R}^{k \times q}$ et vecteur $w \in \mathbb{R}^{(n-q) \times 1}$.
3. Résoudre le problème des moindres carrés :

$$\min_{x \in \mathbb{R}^k} \|Ax - w\|^2$$

4. Imputer par :

$$u = Bx = B\left(A^+\right) w$$

Avantage : non-paramétrique, capture non-linéarités locales.

SVD – Décomposition en valeurs singulières (cas simple)

SVD (Singular Value Decomposition) – cas suffisant de données complètes

1. Séparer données en Y_c (observations complètes) et Y_m (observations incomplètes).
2. Calculer la SVD tronquée de rang J sur Y_c :

$$\hat{Y}_c^{(J)} = U_J D_J V_J^\top$$

3. Estimer coefficients par régression sur composantes observées :

$$\hat{\beta} = (V_J^\top V_J)^{-1} V_J^\top Y_i$$

4. Prédire manquants par :

$$\hat{Y}_i = V_J^{(c)} \hat{\beta}$$

SVD – Complétion itérative (trop de manquants)

SVD itérative (quand $|M|$ est important)

1. Initialiser $Y^{(0)}$ en remplaçant manquants par moyenne.
2. À itération t, résoudre :
$$\min_{U_J, V_J, D_J} \left\| Y - m - U_J D_J V_J^\top \right\|^2$$
(ignorer manquants), puis construire $Y^{(t+1)}$.
3. Répéter jusqu'à convergence :
$$\frac{\|Y^{(t+1)} - Y^{(t)}\|}{\|Y^{(t)}\|} < \varepsilon \quad (\varepsilon = 10^{-6})$$

Avantage : gère forte proportion de manquants.

MICE (Multivariate Imputation by Chained Equations)

Idée : imputation itérative par chaînes de régressions (une régression par variable avec manquants).

1. Initialiser manquants par moyenne.
2. Pour chaque variable Y_j avec manquants :
   - Régresser Y_j sur toutes les autres variables Y_{-j} (à partir des dernières imputations).
   - Imputer manquants de Y_j par prédiction + bruit aléatoire (pour varibilité).
3. Répéter m fois (typiquement $m = 5$ imputations multiples).
4. **Analyse :** analyser chaque dataset imputé séparément, combiner résultats par règles de Rubin.

Avantages : robuste, flexible (fonctionne bien sous MAR), produit imputation multiple.
Limite : coûteux en calcul, sensible à l'ordre des variables (moins critique avec assez d'itérations).

Imputation générative – fondements bayésiens

Base : inférence bayésienne

$$p(\theta \mid Y_{obs}) \propto p(\theta)\, f(Y_{obs}; \theta)$$

Data augmentation (Tanner & Wong, 1987)

- **I-step :** échantillonner $Y_{mis}^{(t+1)} \sim p(Y_{mis} \mid Y_{obs}, \theta^{(t)})$.
- **P-step :** échantillonner $\theta^{(t+1)} \sim p(\theta \mid Y_{obs}, Y_{mis}^{(t+1)})$.

Exemple : Amelia II (Honaker et al., 2011)

- Hypothèses : $Y \sim \mathcal{N}_k(\mu, \Sigma)$ avec MAR.
- Paramètres : $\theta = (\mu, \Sigma)$; estimer via itérations.
- Produit imputations multiples conformes à la distribution jointe.

GAIN (Generative Adversarial Imputation Network)

GAIN (Yoon et al., 2018) – approche GAN

1. Construire entrée bruitée $\bar{X}$ en remplaçant manquants selon masque M.
2. **Générateur :** produire imputation $\hat{X} = G(\bar{X}, M)$.
3. **Discriminateur :** prédire masque $\hat{M} = D(\hat{X})$ (observé vs imputé, variable par variable).
4. Optimiser D par entropie croisée binaire sur M.
5. Optimiser G pour tromper D + terme de reconstruction masqué :

$$\min_{\theta_G} \sum_{i \in B} L_G(M_i, \hat{M}_i) + \alpha \, L_{rec}(\bar{X}_i, \hat{X}_i, M_i)$$

6. Itérer jusqu'à convergence.

Avantage : capture dépendances complexes. **Limite :** training instable (GAN).

VAE Iterative Imputation

Variational Autoencoder pour imputation itérative

1. **Entrées :** $\bar{X}_{train}, M_{train}, \bar{X}_{test}, M_{test}, i_{max}, e_{min}$.
2. Entraîner VAE : $\mathrm{VAE} = \mathrm{train}(\bar{X}_{train}, M_{train})$.
3. Initialiser bruit $\epsilon \sim \mathcal{N}(0,1)$ et

$$\hat{X}_{test} = \bar{X}_{test} \odot M_{test} + \epsilon \odot (1 - M_{test}).$$

4. Itérer jusqu'à convergence ($i \geq i_{max}$ ou $e \leq e_{min}$) :

$$\begin{aligned} \hat{X}_{test} &\leftarrow \mathrm{VAE}(\hat{X}_{test}) \\ e &\leftarrow L_{rec}(\bar{X}_{test}, \hat{X}_{test}, M_{test}) \\ \hat{X}_{test} &\leftarrow \bar{X}_{test} \odot M_{test} + \hat{X}_{test} \odot (1 - M_{test}) \end{aligned}$$

Avantage : itératif, apprend représentation latente. **Limite :** convergence peut être lente.

Universal Marginalizer (one-shot imputation)

Universal Marginalizer – imputation directe (Entraînement)

1. **Entrées :** jeu complet $\mathcal{D}$, réseau f_θ, nombre itérations N_{iter}.
2. Pour $iter = 1, \ldots, N_{iter}$: échantillonner batch complet $X \sim \mathcal{D}$.
3. Échantillonner masque aléatoire $p \sim \mathcal{U}(0,1)$, $M \sim \text{Bernoulli}(p)$.
4. Créer $X_{in} = X \odot M$ et concaténer $\tilde{X} = [X_{in}, M]$.
5. Prédire : $\hat{X} = f_\theta(\tilde{X})$.
6. Perte sur entrées masquées :

$$\mathcal{L} = \frac{1}{\sum(1-M)} \sum (1-M) \cdot \text{Loss}(X, \hat{X})$$

7. Mettre à jour θ par SGD.

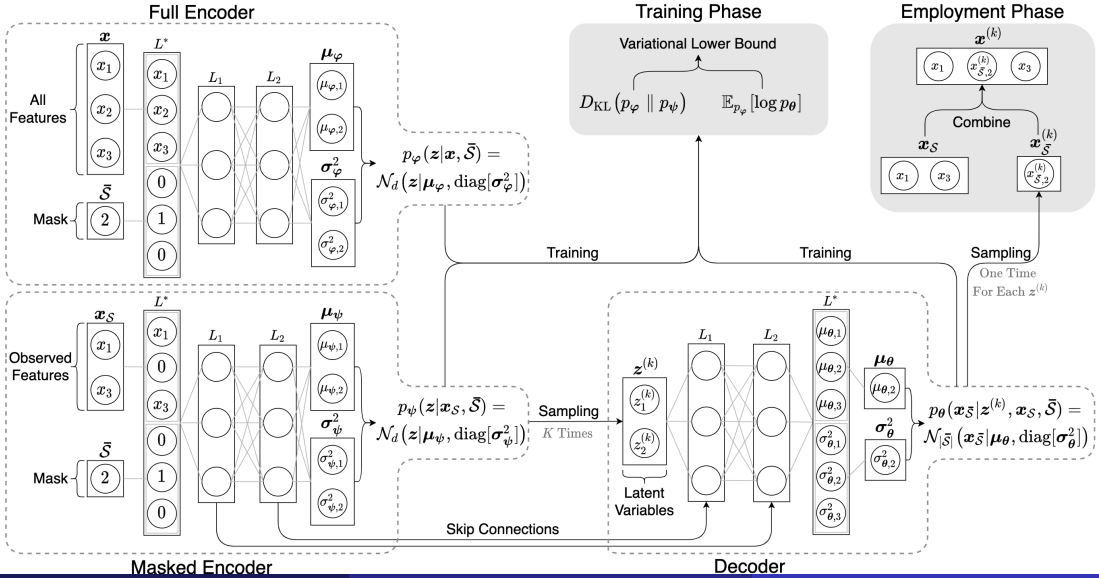
Avantage : one-shot (pas d'itération en inférence). **Limite :** masques variables peuvent être difficiles.

VAEAC – Variational Autoencoder with Arbitrary Conditioning

Idée clé : entraîner un VAEAC pour prédire **n'importe quel sous-ensemble manquant** conditionnellement aux variables observées.

- Le masque M est intégré à l'apprentissage : modèle apprend directement $p(X_{mis} \mid X_{obs}, M)$.
- **Architecture :**
  - Encoder complet sur toutes les features $\Rightarrow$ représentation latente z.
  - Encoder masqué sur seulement features observées.
  - Décodeur produit distribution sur manquants.
- **Avantages :** très adapté données interdépendantes, performant sur données tabulaires, flexible.
- **Limite majeure :** complexité augmente vite avec dimension ("malédiction dimensionnelle") $\Rightarrow$ dégradation stabilité, coût calcul, qualité en très haute dimension.

Architecture du VAEAC (schéma Olsen et al., 2023)



VAE vs VAEAC – comparaison rapide

Aspect	**VAE itératif**	**VAEAC**
Flexibilité masque	Fixe à l'entraînement	Arbitraire
$p(X_{mis}\|X_{obs})$	Approximée itérativement	Directement apprise
Scalabilité dimension	Bonne	Moyenne (malédiction dim.)
Données tabulaires	Acceptable	Très bon
Inférence	Itératif	One-shot
Imputation multiple	Facile (relancer)	Facile (relancer)

Évaluation de la qualité d'imputation

Sur données complètes (masquage synthétique) :

- **RMSE :** $\sqrt{\frac{1}{|\text{manquants}|}\sum(X_{vrai} - X_{imput})^2}$ – sensible aux outliers.
- **MAE :** erreur absolue moyenne – plus robuste.
- **Corrélation inter-variables :** préservation structure de dépendance.
- **Distribution marginale :** tester KS-test, Wasserstein, etc.

Sur données réelles avec manquants (indirect) :

- **Performance downstream :** AUC, F1, prédiction après imputation.
- **Stabilité Shapley :** variance explications avant/après imputation.
- **Analyse de sensibilité :** tester plusieurs méthodes, comparer résultats.
- **Cross-validation :** masquer synthétiquement, évaluer RMSE.

Protocole Expérimental

Données

- 2 datasets médicaux : **Breast Cancer**, **Diabetes**.
- 1 dataset simulé haute dimension :

$$n = 800,\ p = 120$$

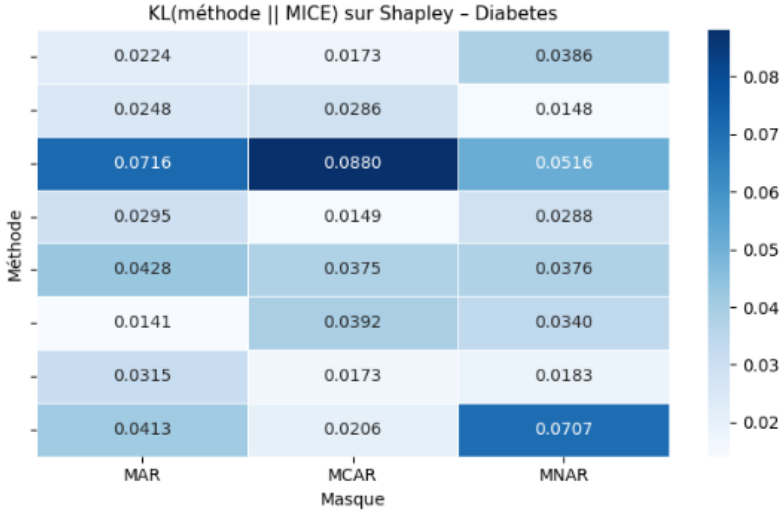
Simulation des manques

- Injection de **MCAR**, **MAR**, **MNAR**.
- Taux de manque : de **5%** à **70%**

Évaluation

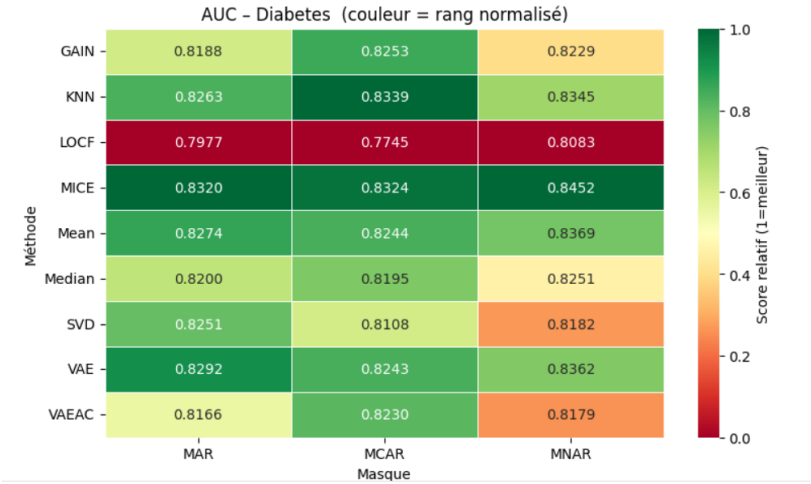
- **9** méthodes comparées.
- Métriques : **RMSE**, **AUC**, **Temps**, **SHAP**.

Implémentation : entraînement du modèle



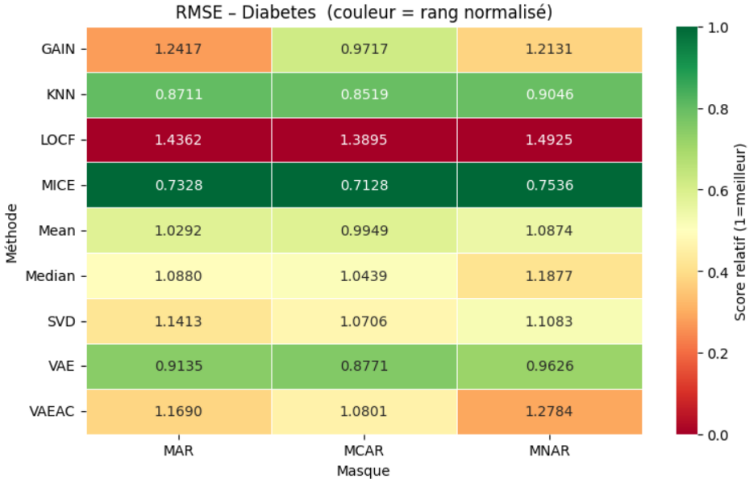
Suivi de la reconstruction et de la régularisation KL pendant l'entraînement (dataset Diabetes).

Analyse RMSE (MCAR)



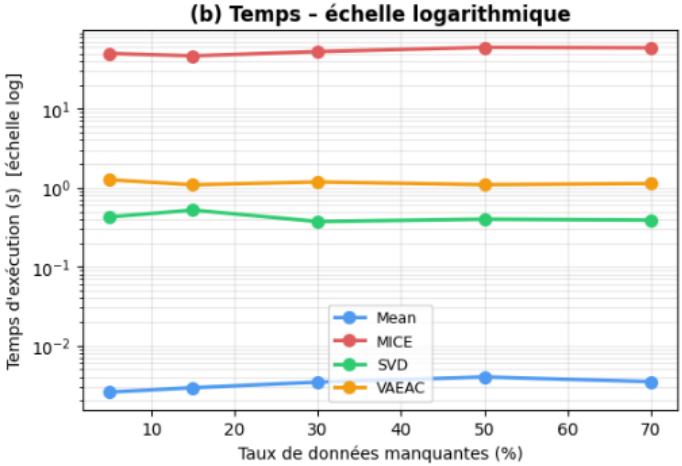
Sous MCAR, MICE/KNN dominent ; les imputations simples restent moins robustes.

Analyse RMSE (MNAR)



Sous MNAR, la moyenne/médiane se dégrade nettement davantage.

Crash test de scalabilité (temps d'exécution)



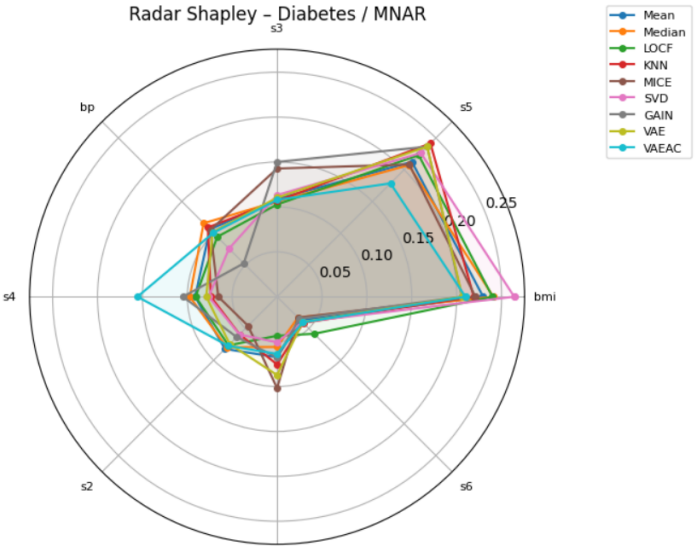
MICE augmente rapidement en grande dimension ; VAEAC reste plus stable à l'inférence.

Analyse SHAP : C'est quoi une valeur de Shapley ?

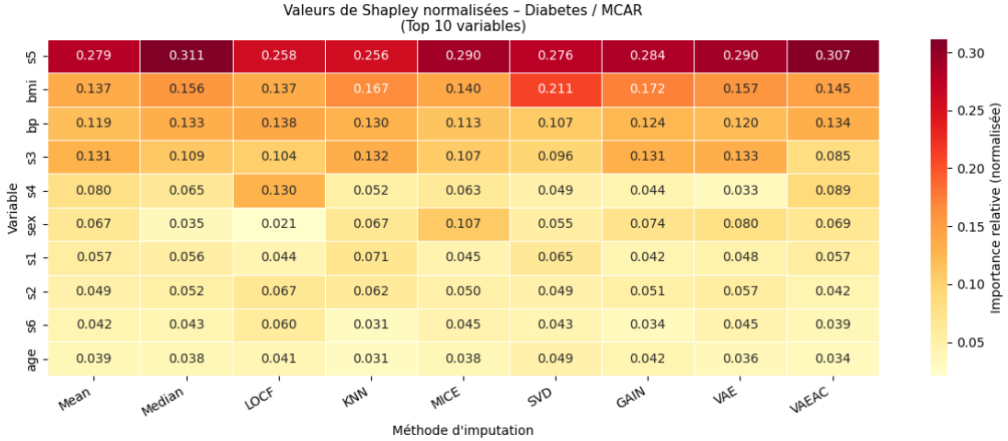
$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M - |S| - 1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)]$$

Méthode utilisée dans l'expérience : TreeSHAP (calcul exact/rapide des valeurs de Shapley pour modèles d'arbres).

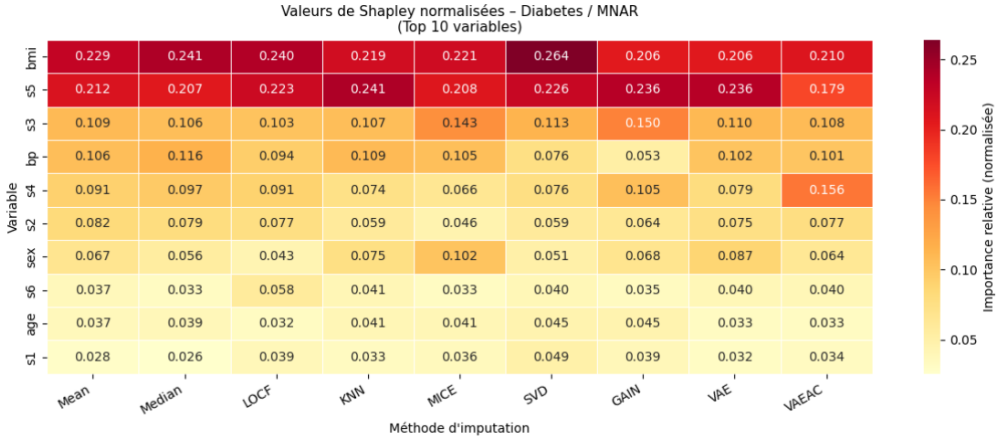
Analyse SHAP : vue d'ensemble



Analyse SHAP sous MCAR



Analyse SHAP sous MNAR

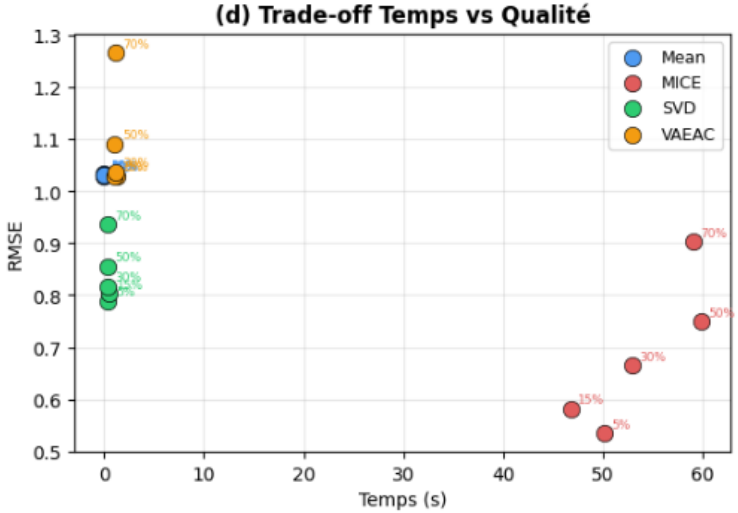


Reproducibilité des résultats via notebook disponible sur le repo github

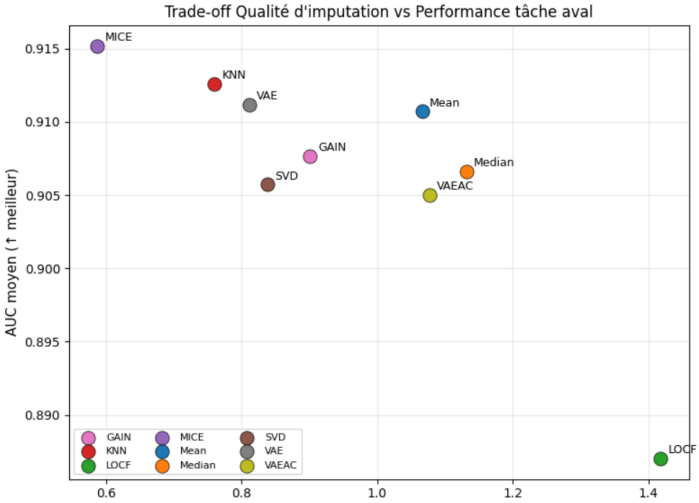
Conséquences d'une mauvaise imputation

- **Baisse de précision prédictive :** une imputation inadéquate dégrade la qualité des variables d'entrée et réduit la performance des modèles.
- **Introduction de biais :** des hypothèses d'imputation incorrectes peuvent déformer les distributions et amplifier des biais statistiques.
- **Impact sur l'explicabilité :** les explications locales peuvent devenir instables ou trompeuses lorsque les données imputées ne respectent pas la structure réelle.
- **Shapley values :** une mauvaise utilisation de l'imputation (notamment un **VAEAC à prior gaussien**) peut biaiser le calcul des valeurs de Shapley et mener à des explications biaisées.

Trade-off : classement temps/performance



Trade-off global qualité–temps



Arbre décisionnel – Quelle méthode choisir ?

Guide empirique par contexte :

- $|M| < 5\%$ **ET MCAR :**
  - $\rightarrow$ **Suppression** (listwise ou pairwise). Simple, pas de biais.
- $5\% < |M| < 30\%$ **ET MAR ET données tabulaires :**
  - $\rightarrow$ **MICE** (robuste, imputation multiple) ou **KNN** (rapide).
- $|M| > 30\%$ **ET structure complexe :**
  - $\rightarrow$ **VAEAC** (données tabulaires interdépendantes) ou **GAIN** (données mixtes).
- **MNAR suspecté :**
  - $\rightarrow$ **Analyse de sensibilité** + documentation explicite du mécanisme.
- **Besoin explicabilité (Shapley, feature importance) :**
  - $\rightarrow$ Attention à imputation ! Évaluer stabilité des explications. Utiliser imputation multiple.

Bonnes pratiques

1. **Analyser le mécanisme :** MCAR, MAR ou MNAR ? Documenter.
2. **Choisir méthode appropriée :** ne pas utiliser "une seule" méthode, tester plusieurs.
3. **Évaluation empirique :** si données complètes, masquer synthétiquement et évaluer RMSE/MAE.
4. **Imputation multiple :** produire m versions imputées, analyser séparément, combiner (règles de Rubin).
5. **Sensibilité :** reporter variations dues à imputation dans résultats finaux.
6. **Explicabilité :** vérifier stabilité des explications (Shapley, feature importance) avant/après imputation.
7. **Documentation :** décrire choix d'imputation dans section "Méthodes", rapport technique.

À retenir

- **Les données manquantes ne sont jamais "un détail"** – elles impactent validité, biais et explicabilité.
- **Le choix de méthode dépend du contexte :** volume manquants, mécanisme (MCAR/MAR/MNAR), nature données.
- **Classique n'est pas hors jeu :** MICE, KNN, SVD restent efficaces, robustes et interprétables.
- **Générative pour complexité :** GAIN, VAE, VAEAC gèrent bien structures non-linéaires et interdépendances.
- **Toujours évaluer :** RMSE/MAE, stabilité explicabilité, sensibilité des résultats finaux.
- **Imputation multiple :** produit meilleure incertitude que imputation simple.

Références clés

- Little, R.J.A. & Rubin, D.B. (2002). *Statistical Analysis with Missing Data*. Wiley.
- Rubin, D.B. (1987). *Multiple Imputation for Nonresponse in Surveys*. Wiley.
- Honaker, J., King, G., & Blackwell, M. (2011). Amelia II : A Program for Missing Data. *Journal of Statistical Software*, 45(7).
- Yoon, J., Jordon, J., & van der Schaar, M. (2018). GAIN : Missing Data Imputation using Generative Adversarial Nets. *ICML*.
- Olsen, C. et al. (2023). VAEAC : Variational Autoencoder with Arbitrary Conditioning. *ICLR*.
- Université de Toulouse - Cours sur l'imputation de données. *WikiStats*.
- Understanding Tree SHAP for Simple Models. *shap.readthedocs.io*..

Merci pour votre attention !

Questions ?

Annexe : Résultats complémentaires

Table – Comparaison de la RMSE de reconstruction des valeurs de Shapley (Moyenne $\pm$ Std Dev) sur différents datasets.

Dataset	Mean	Zero	KNN	MissForest	VAEAC
ABALONE	0.7024 ± 0.0618	0.7103 ± 0.0333	1.0832 ± 0.1390	1.1265 ± 0.1168	0.7463 ± 0.0397
CALIFORNIA	0.9695 ± 0.0297	0.8882 ± 0.0448	0.9491 ± 0.0152	0.9113 ± 0.0497	0.9407 ± 0.0035
DIABETES	0.8751 ± 0.0519	0.8833 ± 0.0444	0.8594 ± 0.0310	0.8445 ± 0.0159	0.8902 ± 0.0131
IID	1.0035 ± 0.0082	1.0093 ± 0.0051	1.0087 ± 0.0196	1.0036 ± 0.0034	0.9948 ± 0.0037
GMM	0.8810 ± 0.0322	0.8471 ± 0.0138	0.8493 ± 0.0342	0.8312 ± 0.0055	0.8502 ± 0.0120
COPULA	0.6230 ± 0.0477	0.7388 ± 0.0231	0.6482 ± 0.0320	0.6581 ± 0.0308	0.6922 ± 0.0751

Table – Détail des performances (RMSE) selon la dimension du masque (k).

Dataset	k	Mean	MissForest	VAEAC
COPULA	1	**0.525**	0.748	0.636
	2	0.706	0.796	**0.699**
	4	**0.623**	0.684	0.625
	8	**0.652**	0.680	0.711
GMM	1	**0.653**	0.677	0.666
	2	**0.649**	0.697	0.662
	4	**0.646**	0.689	0.680
	8	0.721	**0.675**	0.727
IID	1	1.154	1.120	**1.106**
	2	1.119	1.148	**1.117**
	4	1.118	**1.097**	1.114
	8	1.091	**1.082**	1.086

Annexe : Schemas complémentaires

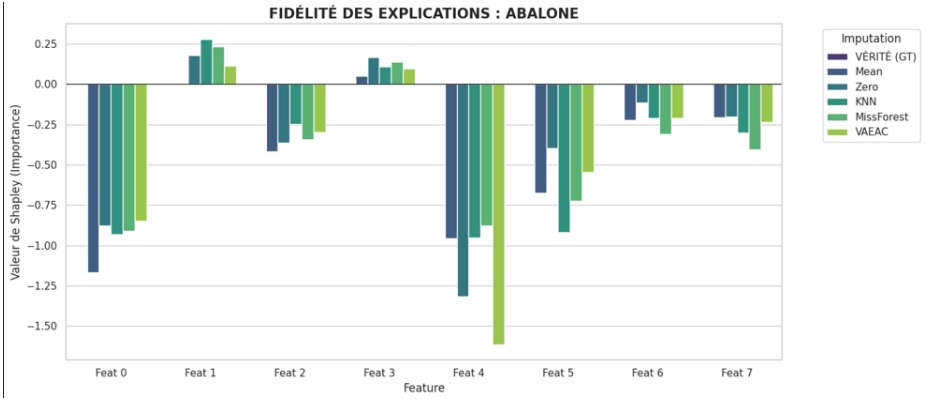


Figure – Évaluation de la fidélité des valeurs de Shapley (Moyenne ± Écart-type) selon la méthode d'imputation sur le dataset *Abalone*. Les résultats sont agrégés sur 3 itérations (*runs*) indépendantes.

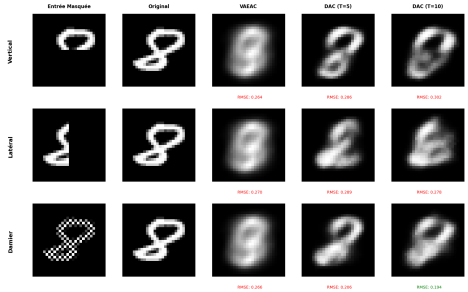


Figure – **Comparaison visuelle et métrique de l'imputation (VAEAC vs DAC).** Reconstruction du chiffre "8" sous trois scénarios de masquage (Vertical, Latéral, Damier). Les résultats comparent le VAEAC classique et le modèle DAC avec $T = 5$ et $T = 10$ pas de diffusion. *Détails expérimentaux :* Entraînement sur **10 époques** avec une taille de lot (*batch size*) de **64**. Ces résultats sont issus d'une **architecture générale légèrement modifiée** et utilisent une **fonction de perte adaptée**. Les scores RMSE (en rouge/vert) indiquent la qualité de reconstruction pixel à pixel.

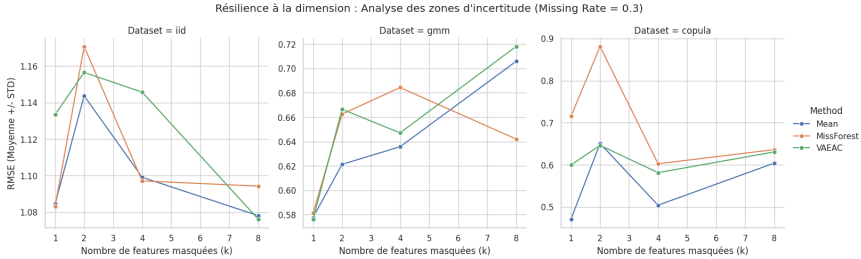


Figure – **Analyse des zones d'incertitude :** Évolution de l'erreur de reconstruction (RMSE) en fonction du nombre de caractéristiques masquées ($k \in \{1, 2, 4, 8\}$) pour les datasets IID, GMM et Copula. Le taux de données manquantes global est fixé à 30%.

Le Problème du Lissage : Bimodalité

Distribution marginale

Mélange équiliséde deux Gaussiennes :

$$p(x) = \frac{1}{2}\mathcal{N}(x; -\mu, \sigma^2) + \frac{1}{2}\mathcal{N}(x; \mu, \sigma^2)$$

Insight clé

La solution analytique : $\hat{x}^* = \mathbb{E}[X] = \frac{1}{2}(-\mu) + \frac{1}{2}(\mu) = \boxed{0}$

Cas Conditionnel : Le Paradoxe du Lissage

Même avec info conditionnelle $x_S = 0$ (au point de symétrie) :

Distribution conditionnelle :

$$p(x_{\bar{S}} \mid x_S = 0) = \sum_{k=1}^{2} \frac{1}{2} \mathcal{N}(x_{\bar{S}}; \hat{\mu}_k(0), \hat{\Sigma})$$

Reste **bimodale** (symétrique)

Mais l'espérance :

$$\mathbb{E}[x_{\bar{S}} \mid x_S = 0] = \frac{1}{2}\hat{\mu}_1 + \frac{1}{2}(-\hat{\mu}_1) = \boxed{0}$$

Perte d'entraînement (décodeur VAEAC) :

$$\hat{x}^* =_{\hat{x}} \mathbb{E}_x[\|x - \hat{x}\|^2]$$