Portfolio : Processus de Datamining

SAE - Mise en œuvre d'un processus de Datamining

Étudiant : Aimad Hamdaoui | **Stack :** Python, Scikit-Learn, Pandas, XGBoost

Ce document présente un projet d'analyse prédictive réalisé dans le cadre de la **SAE "Mise en œuvre d'un processus de Datamining"**. Le but était de construire un modèle capable de prédire si un patient est atteint de diabète à partir de ses constantes de santé. L'objectif de ce portfolio est de justifier la validation de la compétence **ANALYSER**. À travers l'exploration des données, le choix d'une méthodologie d'évaluation rigoureuse et la sélection de métriques adaptées au monde médical, je démontre ma capacité à conduire une analyse scientifique fiable.

ANALYSER : Pertinence des données

Étape 1 : Analyse Exploratoire

Comprendre les données avant de calculer

On ne peut pas construire un bon modèle d'intelligence artificielle sur des données absurdes. J'ai validé ma capacité d'analyse en refusant d'appliquer aveuglément des algorithmes et en étudiant d'abord le sens médical des chiffres :

- **Détection d'anomalies biologiques :** En analysant le fichier de données, j'ai remarqué que certains patients avaient une tension artérielle ou un taux de glucose égal à zéro. Biologiquement, c'est impossible (le patient serait mort). Mon script repère et supprime ces aberrations. Si je n'avais pas fait cette analyse métier, l'IA aurait appris sur de fausses informations.
- **Cartographie des corrélations (Heatmap) :** Avant de lancer la moindre prédiction, j'ai généré une "carte de chaleur". Cet outil visuel m'a permis de comprendre mathématiquement quelles constantes de santé étaient le plus liées entre elles (par exemple, observer que le taux d'insuline monte souvent en même temps que le glucose). Cela permet d'identifier les signaux d'alerte forts pour la maladie.

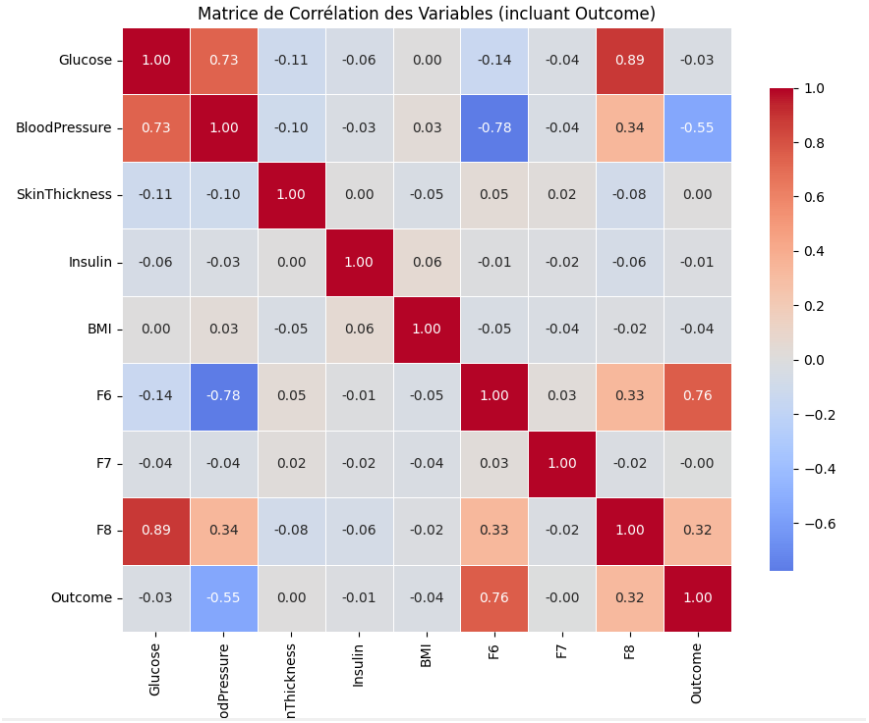

FIG. 1 : Analyse visuelle des corrélations pour identifier les variables influentes.

ANALYSER : Méthodologie d'Évaluation

Étape 2 : Rigueur Scientifique

Garantir des résultats fiables (Validation Croisée)

Dans un processus de Datamining, il est très facile de créer un modèle qui "apprend par cœur" et triche lors de l'évaluation finale. Pour prouver ma maîtrise de l'analyse, j'ai mis en place un système de test extrêmement rigoureux :

- **Le K-Fold (Validation Croisée) :** Au lieu de couper mes patients en un simple groupe d'entraînement et un groupe de test (ce qui laisse la place à la chance), j'ai découpé mes données en 10 morceaux. L'IA s'entraîne sur 9 morceaux et passe son "examen" sur le 10ème, et on répète cela 10 fois en tournant. Cela garantit que l'algorithme est testé de manière fiable.
- **La Stratification pour l'équilibre :** Dans le domaine médical, il y a souvent beaucoup plus de personnes saines que de malades (données déséquilibrées). J'ai configuré mon analyse (mode

  `Stratified`

) pour forcer le système à garder exactement la même proportion de malades et de sains dans chaque test. Sans cette analyse technique, l'algorithme aurait pu ignorer les malades pour gonfler son score de réussite.

```
cv = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)
```

FIG. 2 : Configuration de la Validation Croisée stratifiée (K-Fold).

ANALYSER : Choix des Métriques IA

Étape 3 : Modélisation Métier

Adapter l'algorithme au besoin du médecin

Une bonne analyse de Datamining ne cherche pas seulement à avoir "une bonne note globale", elle cherche à répondre au vrai problème. J'ai prouvé ma capacité d'analyse en adaptant la façon dont la machine est jugée :

- **Choisir la bonne métrique (Le Recall) :** En médecine, dire à un malade qu'il va bien (Faux Négatif) est une erreur grave. Dire à quelqu'un de sain qu'il est malade (Faux Positif) est stressant, mais moins dangereux. C'est pourquoi j'ai programmé l'analyse pour surveiller en priorité le **Recall** (la sensibilité), c'est-à-dire la capacité de l'IA à ne rater aucun malade, plutôt que de regarder uniquement le pourcentage de réussite global.
- **Mise à la même échelle (Pipeline) :** Les algorithmes comme les "Plus Proches Voisins" (KNN) calculent des distances. Or, le glucose se compte en centaines, et l'âge en dizaines. J'ai analysé ce biais et créé un

  `Pipeline`

  avec un outil (

  `StandardScaler`

) qui met toutes les valeurs sur la même échelle de mesure pour éviter qu'une variable n'écrase les autres.
- **Analyse de la stabilité (Écart-type) :** J'ai mis en compétition 6 algorithmes (d'une simple règle logique jusqu'a XGBoost). Plutôt que de regarder uniquement le score moyen, mon script analyse l'**écart-type** de chaque algorithme pour éliminer ceux qui ont eu un coup de chance et privilégier l'IA la plus stable.

RANG	MODÈLE	ACC (%)	SENSIBILITÉ (%)	PRÉCISION (%)	SPÉCIFICITÉ (%)	STD
1	CatBoost	94.80%	94.40%	95.29%	95.18%	+/- 3.12%
2	XGBoost	94.60%	95.20%	94.32%	93.97%	+/- 4.10%
3	Random Forest	94.40%	94.00%	94.80%	94.77%	+/- 4.36%
4	Régression Logistique	89.20%	88.83%	89.84%	89.53%	+/- 3.82%
5	OneR (Arbre 1 règle)	88.40%	86.83%	89.99%	89.93%	+/- 4.63%
6	KPPV (5 voisins)	87.60%	85.63%	89.58%	89.52%	+/- 4.27%
7	ZeroR (Baseline)	49.80%	90.00%	45.00%	10.00%	+/- 0.60%

FIG. 3 : Mise en compétition des algorithmes et analyse de leur stabilité (écart-type).