DIABETES ANALYSIS

A. LARHLIMI
HLAL KAWTAR
École Nationale de Commerce et de Gestion (ENCG) - 4ème Année

7 janvier 2026

Table des matières

1 Le Problème (Business Case)

Dans le domaine médical, le suivi de l'évolution d'une maladie chronique est souvent complexe, car il dépend de nombreux facteurs cliniques et biologiques difficilement interprétables conjointement par un médecin seul. La variabilité inter-patient et la charge de travail élevée peuvent conduire à une sous-estimation ou une surestimation de la gravité réelle de la maladie.

1.1 Objectifs et Enjeux

- **Objectif :** Concevoir un modèle de prédiction de la progression de la maladie (variable cible continue du dataset) à partir des caractéristiques cliniques et biologiques du patient.
- **Enjeu critique :** Une erreur de prédiction n'a pas le même impact clinique :
  - **Surestimation :** Entraîne des traitements lourds et des coûts inutiles.
  - **Sous-estimation :** Retarde la prise en charge, aggravant le pronostic vital.

1.2 Les Données (L'Input)

Le jeu de données contient 442 observations et 10 variables explicatives normalisées :
- **X (Features) :** Âge, sexe, BMI, pression sanguine (bp) et six mesures biologiques (s1 à s6).
- **y (Target) :** Variable continue représentant le score de progression de la maladie.

2 Le Code Python

Voici l'implémentation réalisée pour l'analyse et la modélisation :

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_diabetes
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score

# Chargement des donnees
data = load_diabetes()
df = pd.DataFrame(data.data, columns=data.feature_names)
df['target'] = data.target

# Nettoyage et Imputation
X = df.drop('target', axis=1)
y = df['target']
imputer = SimpleImputer(strategy='mean')
X_clean = pd.DataFrame(imputer.fit_transform(X), columns=X.columns)

# Split Train/Test
X_train, X_test, y_train, y_test = train_test_split(X_clean, y, test_size=0.2,
    random_state=42)

# Modelisation
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluation
y_pred = model.predict(X_test)
print(f"MSE: {mean_squared_error(y_test, y_pred):.2f}")
```

```
32 print(f"R2 Score: {r2_score(y_test, y_pred):.2f}")
```

Listing 1 – Analyse du Diabète avec RandomForestRegressor

3 Analyse Approfondie : Nettoyage (Data Wrangling)

Dans ce projet, des valeurs manquantes (NaN) ont été simulées. La plupart des algorithmes de machine learning reposent sur l'algèbre linéaire ; une seule valeur manquante peut faire échouer les calculs.

3.1 La mécanique de l'imputation

Nous utilisons le `SimpleImputer` :

1. **Apprentissage (fit) :** Calcul de la moyenne μ sur les valeurs observées.
2. **Transformation (transform) :** Remplacement des NaN par μ.

3.2 Le coin de l'expert : Data Leakage

La bonne pratique industrielle consiste à séparer le *Train* et le *Test* **avant** l'imputation pour éviter que des informations du futur (le jeu de test) ne fuient dans l'entraînement.

4 Analyse Approfondie : Exploration (EDA)

L'EDA permet de comprendre la structure statistique. Les features sont centrées-réduites autour de 0 avec un écart-type (σ) d'environ $0,046$.

5 Méthodologie du split (Train/Test)

Nous utilisons un ratio **80/20** (Principe de Pareto) :
— **Train (80%) :** Pour apprendre les motifs.
— **Test (20%) :** Pour évaluer la capacité de généralisation.
— **Random State (42) :** Garantit la reproductibilité des résultats.

6 Focus théorique : Random Forest

6.1 La force du groupe (Bagging)

Le Random Forest construit des centaines d'arbres de décision. Il utilise le **Bootstrapping** (échantillonnage avec remise) et la **Feature Randomness** pour réduire la variance et éviter le sur-apprentissage (overfitting).

7 Analyse Approfondie : Évaluation

En régression, nous n'utilisons pas de matrice de confusion mais des mesures d'écart :
— **MSE (Mean Squared Error) :** $\frac{1}{n}\sum(y_{\text{réel}} - y_{\text{prédit}})^2$
— R^2 **(Coefficient de détermination) :** Proportion de la variance expliquée par le modèle (proche de 1 = excellent).

Conclusion

Ce projet démontre que la qualité d'un modèle dépend de la chaîne de décisions logiques, du nettoyage à l'évaluation, afin de garantir une **prise de décision clinique sûre**.