Projet de Prétraitement des Données : Maladie Cardiaque UCI

Stratégies de nettoyage, normalisation et optimisation pour le Machine Learning

OBJECTIF

Transformer un jeu de données brut et hétérogène en un format optimisé pour la classification prédictive.

MÉTHODOLOGIE

Imputation robuste, encodage sélectif, mise à l'échelle (RobustScaler) et transformations avancées.

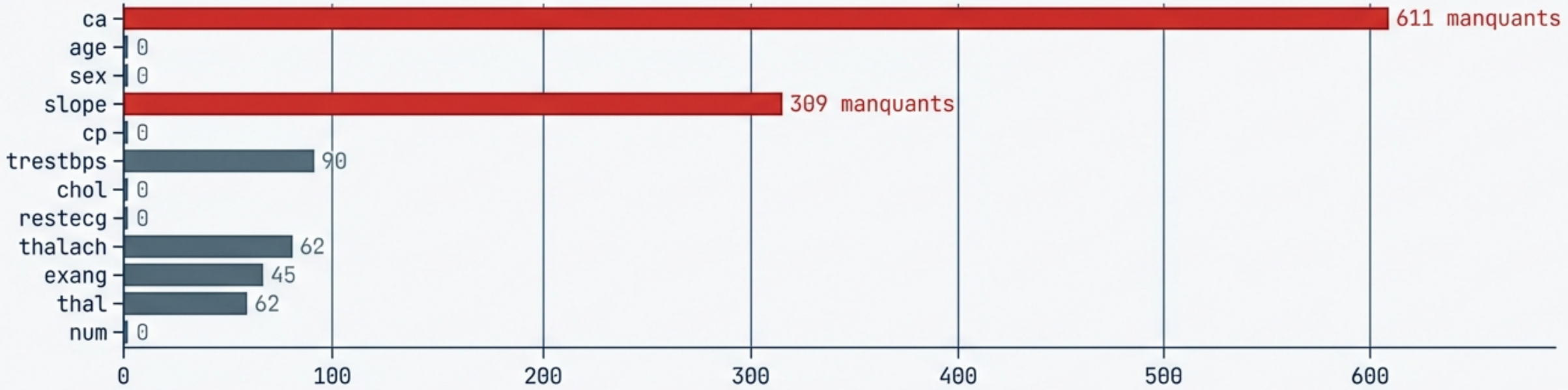
SOURCE

Jeu de données UCI Heart Disease (Kaggle).

Le Diagnostic : Exploration des Données Brutes

920	16	3
Observations	Variables	Types (Float, Int, Object)

Analyse des Valeurs Manquantes (NaN)



Constat critique :
La donnée n'est pas prête. 15 colonnes sur 16 sont **hétérogènes.** Le défi majeur est la gestion des **vides** (NaN) qui **atteignent jusqu'à 66%** pour la variable 'ca'.

Stratégie de Séparation des Variables

Données Brutes

Variables Numériques

Age
Trestbps
Chol
Thalch
Oldpeak

Traitements :
Imputation Statistique,
Discrétisation,
Standardisation

Variables Catégorielles

Sex
Cp
Slope
Thal
Dataset

Traitements :
Imputation Modale,
Encodage (Ordinal vs One-Hot)

Variable Cible (Target)

Nom : 'num' (Classification)

Réparation des Données : Stratégies d'Imputation

STRATÉGIE	DESCRIPTION	CODE / NOTES	STATUT
Option A : Moyenne (Mean)	Sensible aux valeurs extrêmes (Outliers). Risque de distorsion de la distribution.		✚
Option B : KNN Imputer	Précis (basé sur les voisins) mais très coûteux en temps de calcul.		—
Option C : Médiane (Choix Retenu)	Robuste et stable pour les variables numériques physiologiques.	`SimpleImputer(strategy='median')`	✓
Pour Catégories : Mode (Choix Retenu)	Remplace par la valeur la plus fréquente.	`SimpleImputer(strategy='most_frequent')`	✓

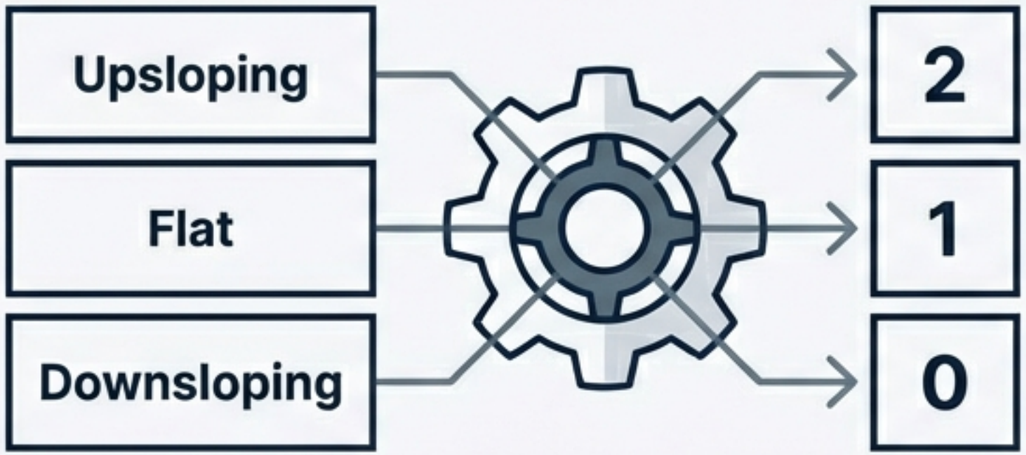
ca	slope
0.0	up
1.0	flat
0.0	up
2.0	down
1.0	flat

Résultat : Zéro valeur manquante.

Traduction pour la Machine : Encodage

1. Encodage Ordinal

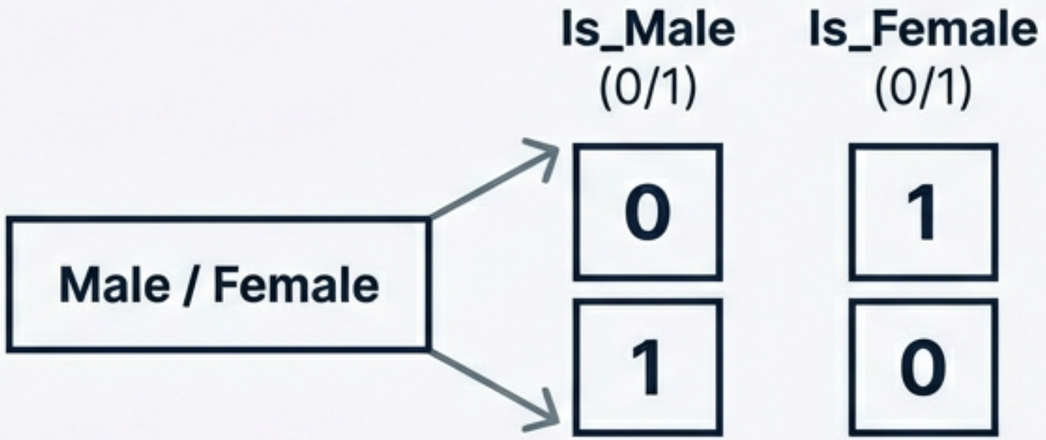
Target Variable: **Slope**



L'ordre hiérarchique est préservé.

2. Encodage Nominal (One-Hot)

Target Variable: **Sex, Chest Pain**

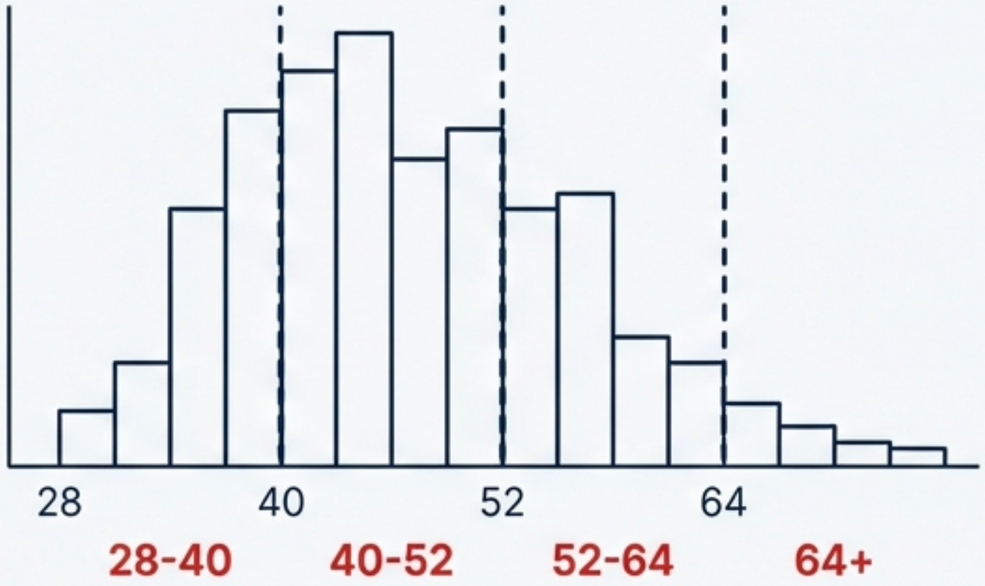



```
OneHotEncoder(drop='first')
```

Pas de hiérarchie. "**drop=first**" élimine la multicolinéarité.

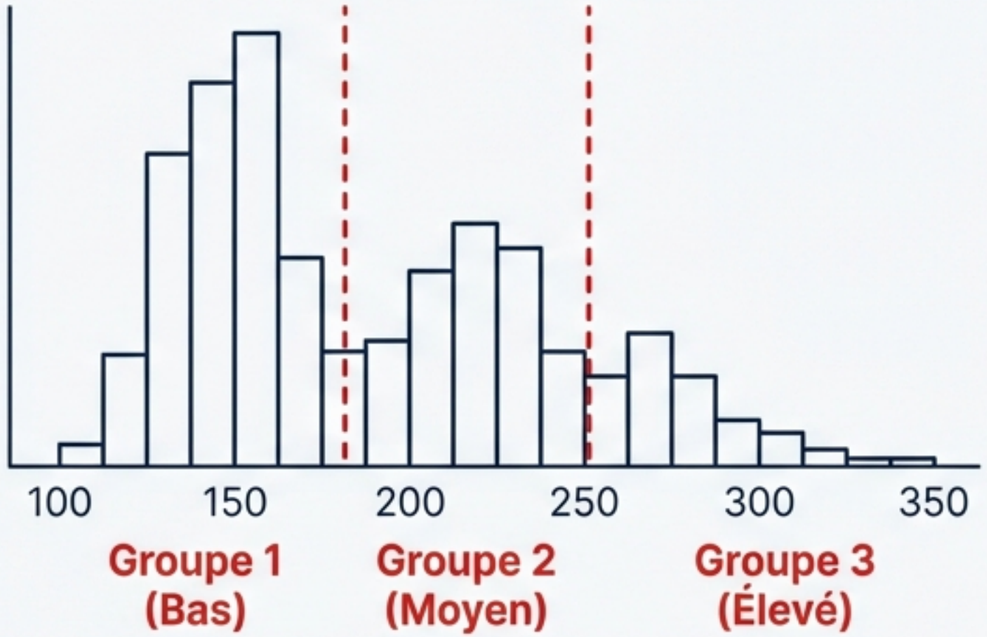
Discrétisation : Créer des Groupes Intelligents

Approche Uniforme (Age)



Découpage arithmétique simple.

Approche KMeans (Cholestérol)



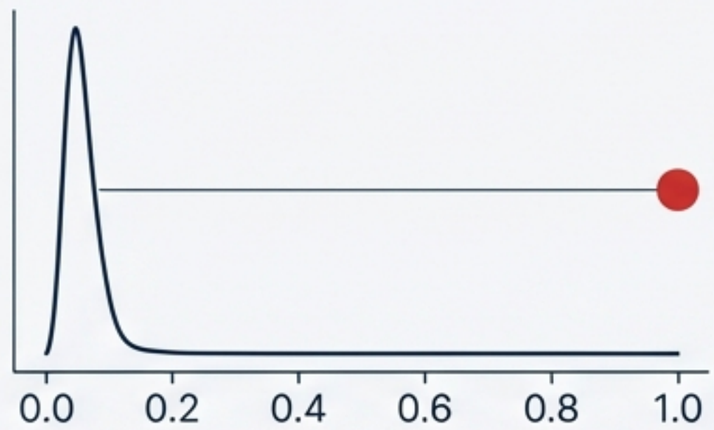
Regroupement par similarité (Clusters).

KMeans capture mieux la structure naturelle des données physiologiques que le découpage uniforme.

La Guerre des Échelles : Normalisation vs Robustesse

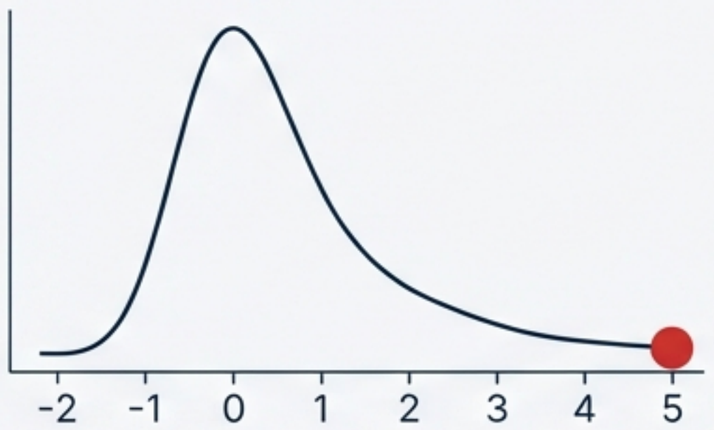
Comparaison des distributions après transformation

MinmaxCaler



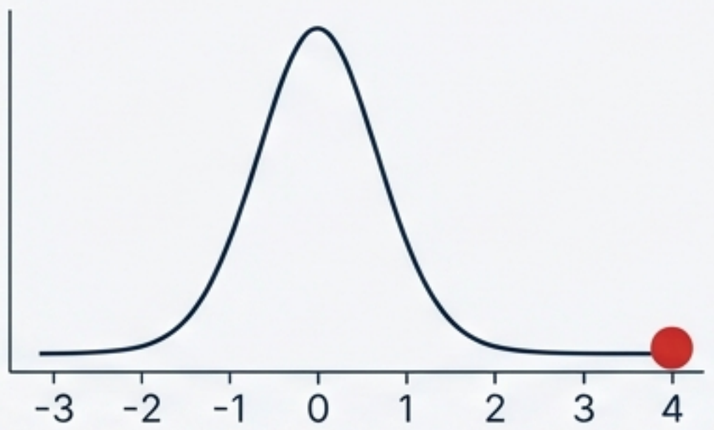
Écrasé par les extrêmes. Intervalle [0, 1].

StandardScaler



Centré (Moyenne=0), mais la variance est gonflée par l'outlier.

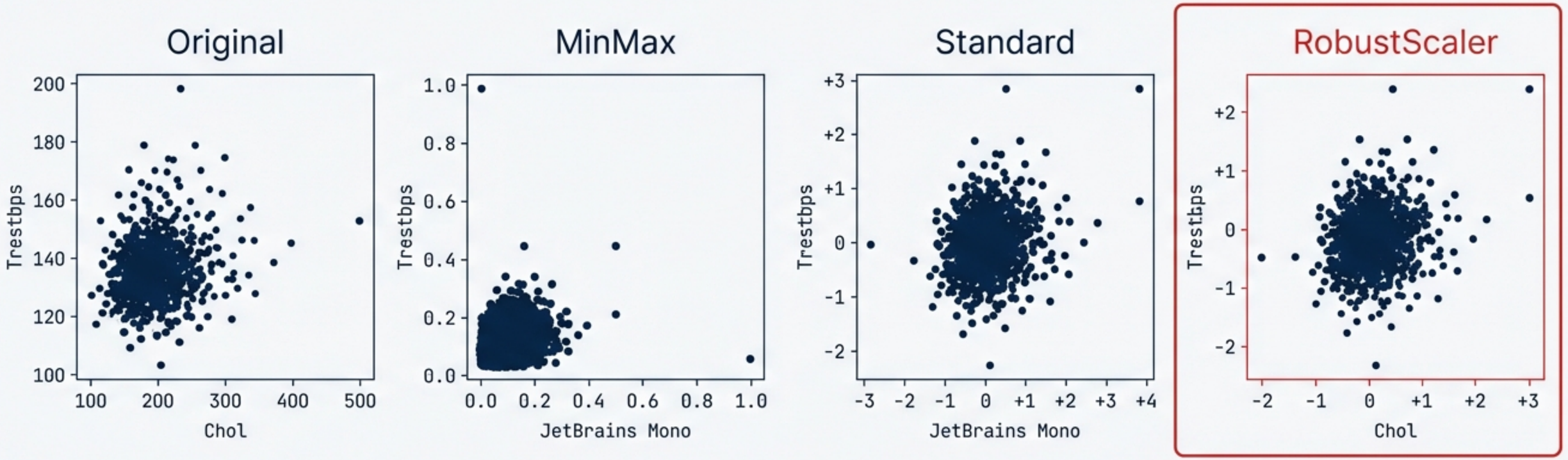
RobustScaler



Centrage sur la Médiane & IQR. Ignore les anomalies.

RobustScaler est indispensable pour les données médicales contenant des valeurs extrêmes.

Visualisation de l'Impact : Trestbps vs Chol



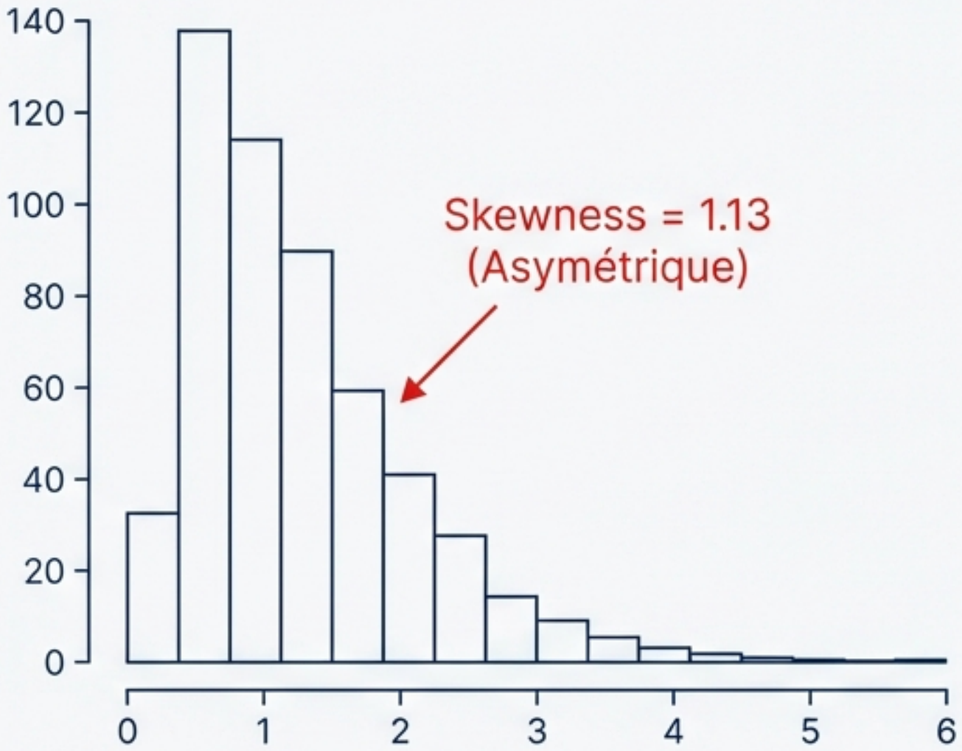
RobustScaler préserve la topologie des données utiles en neutralisant l'effet écrasant des anomalies de cholestérol.

Clinical Data Journalism

Normalisation de la Distribution (PowerTransformer)

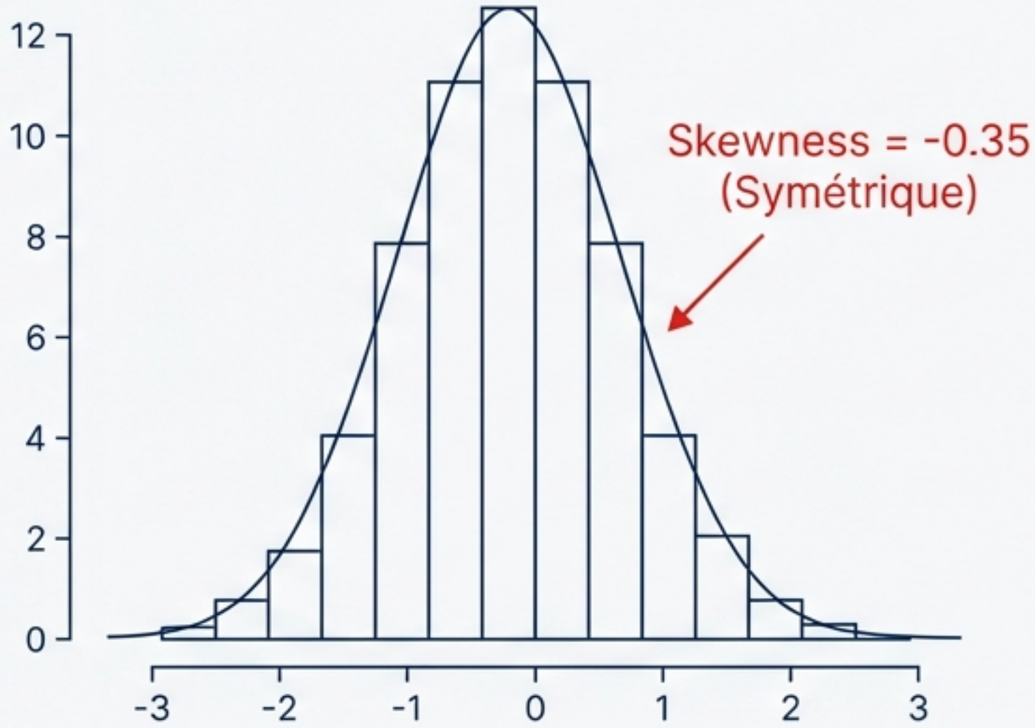
Variable : oldpeak

Avant Transformation



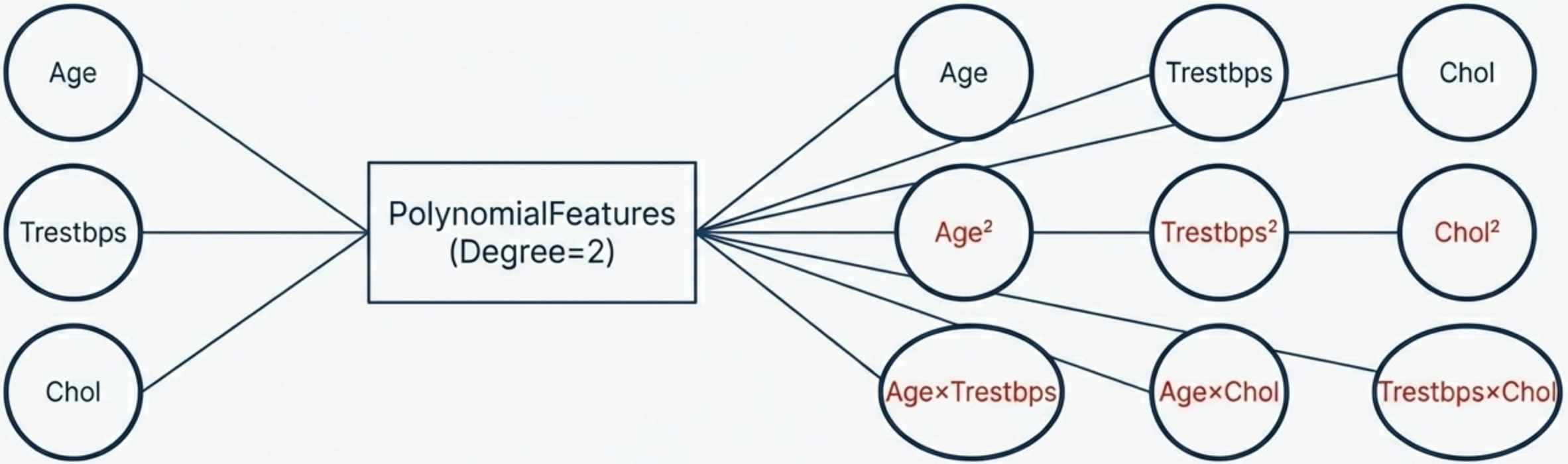
Yeo-Johnson Method

Après Transformation



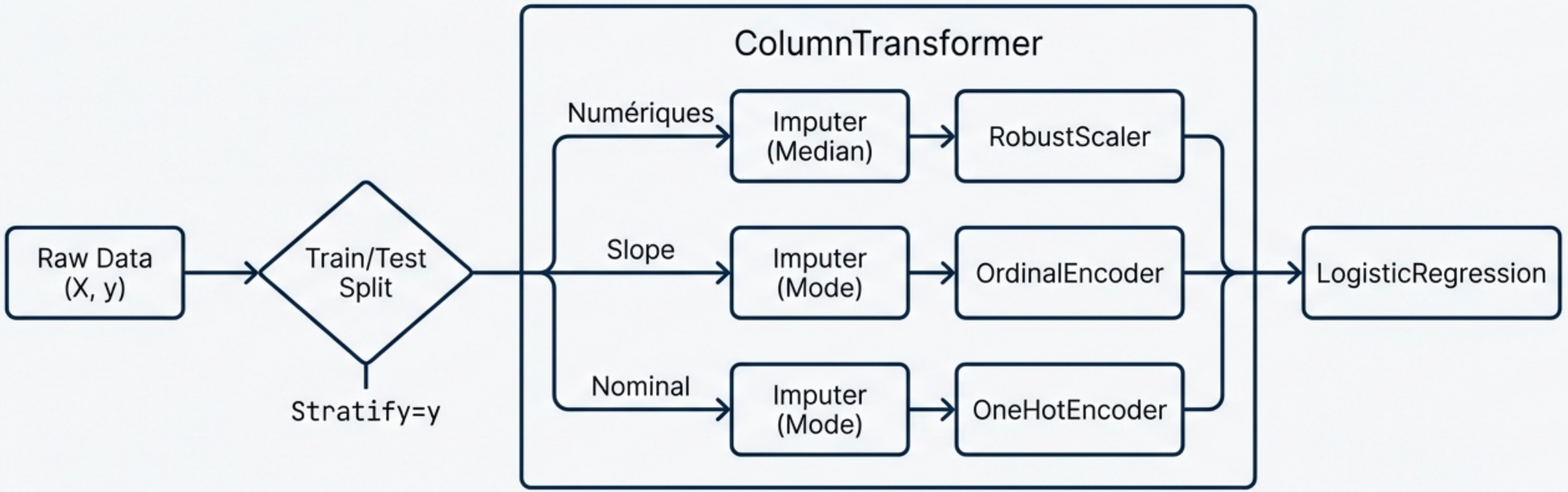
Pourquoi ? De nombreux modèles (comme la Régression Logistique) performent mieux sur des données suivant une loi Normale.

Feature Engineering : Interactions Polynomiales



- **Objectif :** Capturer des relations non-linéaires.
- **Exemple :** Le risque cardiaque peut dépendre de la combinaison 'Age × Cholestérol', même si l'âge seul n'est pas critique.
- **Compromis :** Augmente la dimensionnalité (3 → 9 variables).

L'Architecture du Pipeline Scikit-Learn



L'encapsulation dans un Pipeline garantit l'absence de fuite de données (Data Leakage).

Évaluation de la Performance

Pipeline Standard	Pipeline Robust
59.8%	**60.3%** 
Accuracy	**Accuracy**

Résultat : Léger avantage pour le RobustScaler.

Interprétation : Confirme que la gestion agressive des outliers améliore marginalement la stabilité du modèle linéaire.

Perspective : Pour dépasser 60%, il faudra explorer des modèles non-linéaires (Random Forest, XGBoost).

Conclusions et Bonnes Pratiques

1 **Imputation**

Privilégier la Médiane sur la Moyenne pour les données physiologiques instables.

---

2 **Sémantique de l'Encodage**

Respecter la nature des variables : Ordinal pour l'ordre (Slope), One-Hot pour les catégories (Sex).

---

3 **Scaling**

`RobustScaler` est impératif en présence d'anomalies (outliers). Le `StandardScaler` est risqué sur ce dataset.

---

4 **Architecture**

Utiliser `ColumnTransformer` pour appliquer des traitements différenciés sans briser le pipeline.

---

5 **Distribution**

`Yeo-Johnson` (`PowerTransformer`) est efficace pour corriger les asymétries sévères (Skewness).

Annexe Technique : Définition du Pipeline

```
# Définition des branches de traitement
num_pipe = make_pipeline(SimpleImputer(strategy="median"), RobustScaler())
slope_pipe = make_pipeline(SimpleImputer(strategy="most_frequent"),
                           OrdinalEncoder(categories=slope_cats))
nom_pipe = make_pipeline(SimpleImputer(strategy="most_frequent"),
                         OneHotEncoder(drop="first", handle_unknown="ignore"))

# Assemblage dans le ColumnTransformer
preprocessor = ColumnTransformer(
    transformers=[
        ("num", num_pipe, numeric_features),
        ("slope", slope_pipe, ["slope"]),
        ("nominal", nom_pipe, nominal_cols)
    ]
)

# Pipeline Final
pipeline = make_pipeline(preprocessor, LogisticRegression())
pipeline.fit(X_train, y_train)
```

Code prêt pour la mise en production via Scikit-Learn.

Merci.

Questions & Réponses

Analyse réalisée sur le jeu de données UCI Heart Disease.