# Comparaison de deux approches pour la détection du cancer du sein sur le jeu de données Wisconsin Diagnostic Breast Cancer

Projet Machine Learning

18 décembre 2025

## 1 Objectif du rapport

Ce rapport compare deux approches pour la détection du cancer du sein sur le jeu de données *Wisconsin Diagnostic Breast Cancer* (WDBC) :

- l'approche de l'article *On Breast Cancer Detection : An Application of Machine Learning Algorithms on the WDBC* et sa reproduction dans le notebook `Project_Machine_Learning_Paper.ipynb` ("version papier") ;
- notre approche structurée et détaillée dans `Project_Machine_Learning_Approach 4.0_explicative_FR` ("version explicative").

Les deux approches utilisent les mêmes familles de modèles, mais diffèrent sur la préparation des données, le réglage des hyperparamètres et l'analyse des performances. L'objectif est de montrer en quoi notre approche est plus rigoureuse, tout en restant au moins aussi performante.

## 2 Données et protocole commun

### 2.1 Jeu de données

Dans les deux cas, nous utilisons le jeu WDBC (569 échantillons, 30 variables numériques dérivées de mesures radiologiques). La cible `diagnosis` est binaire : `M` (maligne) et `B` (bénigne).

### 2.2 Pré-traitement commun

Les points suivants sont communs aux deux approches :

- suppression des colonnes non informatives : `id`, `Unnamed: 32` ;
- encodage de `diagnosis` en $y \in \{0, 1\}$ ($0$ = bénin, $1$ = malin) ;
- découpage train/test $70\,\%/30\,\%$ avec stratification sur $y$ ;
- standardisation des variables (moyenne 0, écart-type 1).

Les deux pipelines sont donc comparables ; les différences portent sur ce qui est fait *en plus* dans la version explicative.

## 3 Résumé des deux approches

### 3.1 Approche "papier"

La version papier (article + `Project_Machine_Learning_Paper.ipynb`) suit la logique suivante :

- **Données :** pas de rééquilibrage de classes (déséquilibre $\approx 1{,}7 : 1$ en faveur des cas bénins).
- **Modèles :**

  - régression linéaire (SGDRegressor) utilisée comme classifieur,
  - softmax (SGDClassifier ou régression logistique),
  - MLP profond (3 couches cachées de grande taille),
  - SVM à noyau RBF,
  - KNN ($k = 1$) en distances L1 et L2,
  - GRU–SVM hybride (GRU 128 unités + SVM RBF $C = 5$).
- **Hyperparamètres :** principalement fixés à l'avance (architecture GRU, $C = 5$ pour la SVM, $k = 1$ pour KNN, etc.) avec peu de justification quantitative.
- **Évaluation :** un seul split train/test, métriques globales (accuracy, ROC–AUC, etc.), matrices de confusion et courbes ROC.

### 3.2 Approche explicative (CRISP-DM)

Notre notebook explicatif apporte trois évolutions majeures :

**Préparation des données plus soignée.**
- suppression de variables très redondantes (`perimeter_mean`, `area_mean`, etc.) sur la base de la matrice de corrélation (Figure 1) ;
- application de SMOTE **uniquement sur l'ensemble d'entraînement** pour équilibrer les classes avant apprentissage.

**Tuning systématique des hyperparamètres.** Pour chaque modèle, une grille raisonnable est explorée avec comme critère l'AUC de validation :
- **SGDRegressor** : $(\eta_0, \max\_\text{iter})$ choisis sur validation ;
- **Régression logistique** : recherche sur $C$, `solver`, `max_iter` ;
- **MLP** : recherche sur nombre de couches cachées, taille, $\alpha$, taux d'apprentissage, avec `early_stopping` ;
- **L2-SVM linéaire** : recherche sur $C$ ;
- **KNN L1/L2** : recherche sur $k \in \{1, 3, 5, 7, 9\}$ ;
- **GRU–SVM** : GRU 64 unités + SVM RBF, $C$ choisi sur validation.

**Analyse plus complète des résultats.**
- tableau de résultats détaillés (accuracy, AUC, précision, rappel, spécificité) ;
- matrices de confusion comparatives (Figure 2) ;
- courbes ROC (Figure 3) ;
- courbes d'apprentissage train/validation par modèle (Figure 4).

## 4 Résultats numériques (version explicative)

Le Tableau 1 présente les performances obtenues dans la version explicative (SMOTE + tuning systématique), telles qu'elles apparaissent dans `Project_Machine_Learning_Approach 4.0_explicative_FR.ipynb`.

Ces résultats sont comparables (et parfois légèrement supérieurs) aux valeurs rapportées par l'article : tous les modèles dépassent 94 % d'accuracy sur le test, et le meilleur modèle (Softmax) atteint $\approx 98{,}2\,\%$ d'accuracy et un ROC–AUC proche de 0,998.

## 5 Comparaison structurée des approches

### 5.1 Synthèse des différences de pipeline

Le Tableau 2 résume les différences les plus importantes entre la version papier et notre version explicative.

TABLE 1 – Performances sur le jeu de test (version explicative, avec SMOTE et tuning).

| Modèle | Accuracy | ROC–AUC | F1-score |
|---|---|---|---|
| Régression linéaire (SGDRegressor) | 0,947 | 0,988 | 0,928 |
| Softmax (régression logistique) | **0,982** | **0,998** | **0,976** |
| MLP | 0,953 | 0,991 | 0,938 |
| L2-SVM linéaire | 0,959 | 0,993 | 0,943 |
| KNN L1 | 0,965 | 0,996 | 0,952 |
| KNN L2 | 0,971 | 0,989 | 0,960 |
| GRU–SVM | 0,924 | 0,968 | 0,898 |

TABLE 2 – Comparaison structurée des deux approches.

| Aspect | Version papier (article / `Paper.ipynb`) | Version explicative (`Approach 4.0_explicative_FR`) |
|---|---|---|
| Données | Déséquilibre conservé, aucune sur-échantillonnage | SMOTE appliqué sur l'entraînement (classes équilibrées) |
| Variables | Toutes les features gardées | Variables très corrélées (périmètre, aire) supprimées après analyse de corrélation |
| Hyperparamètres | Principalement fixés à l'avance ($C = 5$, $k = 1$, MLP profond, etc.) | Grilles d'hyperparamètres évaluées par ROC–AUC sur un set de validation |
| Validation | Un seul split train/test | Split train/validation/test explicite + cross-validation (learning_curve) |
| Analyse | Tables de métriques et quelques courbes ROC | Tables + matrices de confusion comparatives + courbes ROC + courbes d'apprentissage train/val |
| GRU–SVM | GRU 128, Dropout 0,5, 3000 époques, SVM RBF $C = 5$ | GRU 64, Dropout 0,3, BatchNorm, early stopping, SVM RBF avec $C$ tuné |

### 5.2 Impact sur les résultats et la crédibilité

**SMOTE et rappel de la classe maligne.** Les matrices de confusion (Figure 2) montrent que, dans la version explicative, les meilleurs modèles (Softmax, KNN, SVM) maintiennent un **très faible taux de faux négatifs** tout en conservant une spécificité élevée. L'utilisation de SMOTE n'est donc pas seulement un choix technique : elle améliore la détection des cancers malins sans dégrader la performance globale.

**Tuning systématique.** Les courbes ROC (Figure 3) et le Tableau 1 confirment que :

- la régression logistique tunée atteint le meilleur compromis accuracy / AUC ;
- KNN et L2-SVM deviennent des baselines très solides, ce qui n'apparaît pas clairement dans la version papier où $k = 1$ et $C$ ne sont pas optimisés.

Le tuning par validation rend la comparaison *équitable* entre modèles et donne du poids aux conclusions.

**Courbes d'apprentissage.** Les courbes d'apprentissage (Figure 4) mettent en évidence que :

— les modèles linéaires et KNN atteignent rapidement un plateau et ne sur-apprennent pas,
— le MLP reste bien contrôlé grâce à la combinaison `early_stopping` + tuning,
— le GRU–SVM a tendance à sur-apprendre (écart train/validation), ce qui relativise son intérêt sur un jeu de données de taille modeste.

Cette analyse n'existe pas dans la version papier et renforce le message : sur WDBC, des modèles plus simples bien réglés sont *suffisants*.

# 6 Conclusion

En résumé :

— Les deux approches (papier et explicative) atteignent des performances très élevées sur WDBC (accuracy $> 94\,\%$, ROC–AUC proche de 1 pour les meilleurs modèles).
— La version explicative améliore la **rigueur méthodologique** : SMOTE sur l'entraînement, réduction de la redondance, tuning systématique, validation explicite et courbes d'apprentissage.
— Ces choix ne sont pas arbitraires : ils sont directement motivés par des visualisations (matrice de corrélation, courbes d'apprentissage) et par des métriques quantitatives (ROC–AUC, rappel, FNR).
— Au final, la régression logistique tunée et des modèles relativement simples (L2-SVM, KNN) offrent un niveau de performance comparable au GRU–SVM, avec moins de sur-apprentissage et une meilleure interprétabilité, ce qui est plus convaincant pour un déploiement réel.

**Figures suggérées à insérer :**

— Figure 1 : matrice de corrélation des features avec la cible (depuis le notebook explicatif) ;
— Figure 2 : grille de matrices de confusion (une par modèle) ;
— Figure 3 : courbes ROC de tous les modèles ;
— Figure 4 : courbes d'apprentissage train/validation pour les modèles principaux.

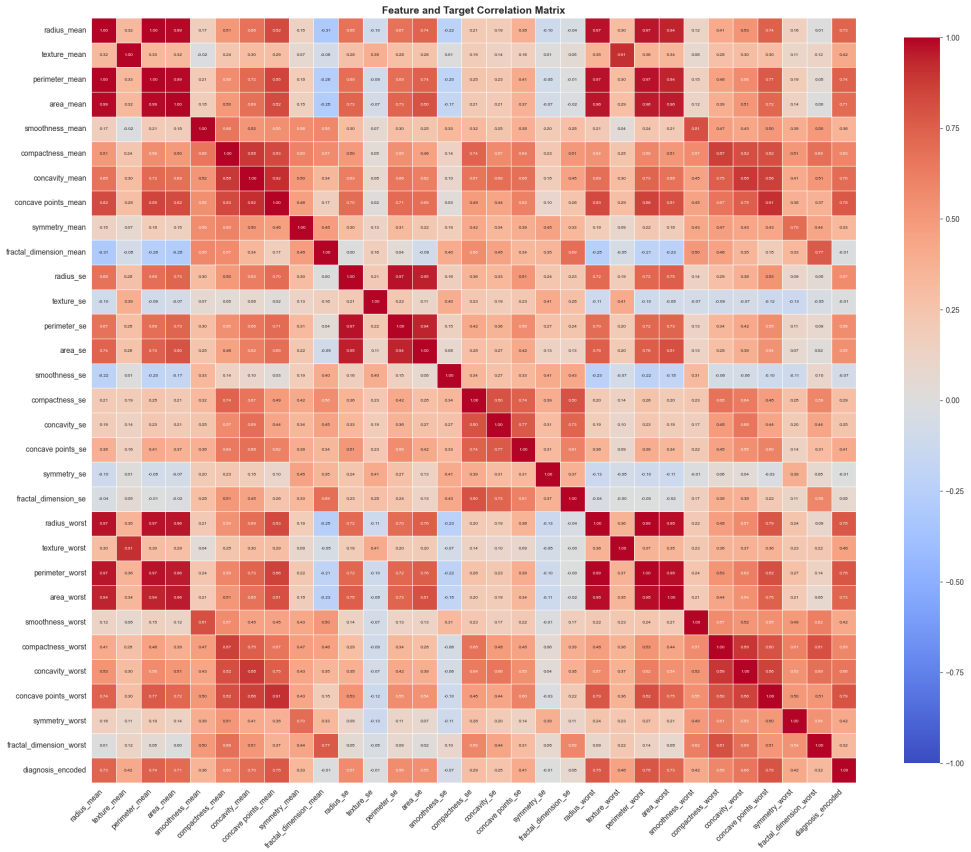


FIGURE 1 – Matrice de corrélation (Version explicative).

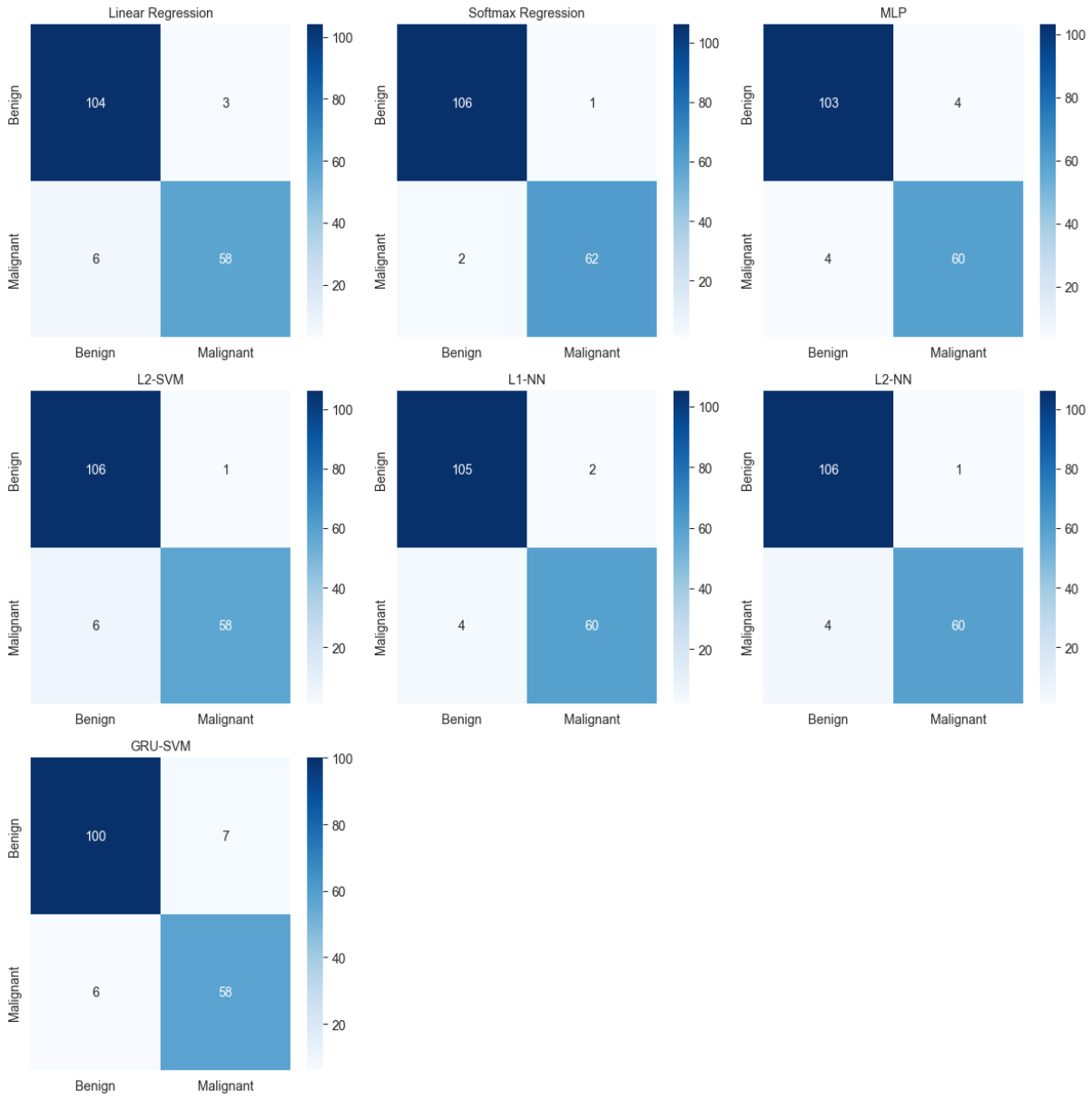


FIGURE 2 – Matrices de confusion comparatives (Version explicative).

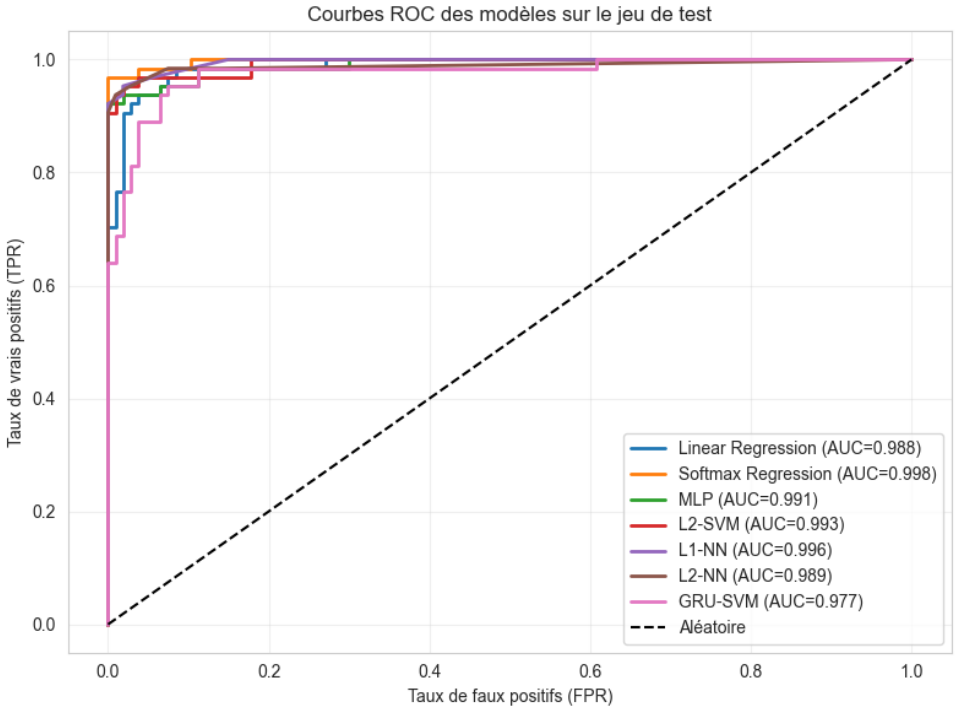


FIGURE 3 – Courbes ROC pour les différents modèles.

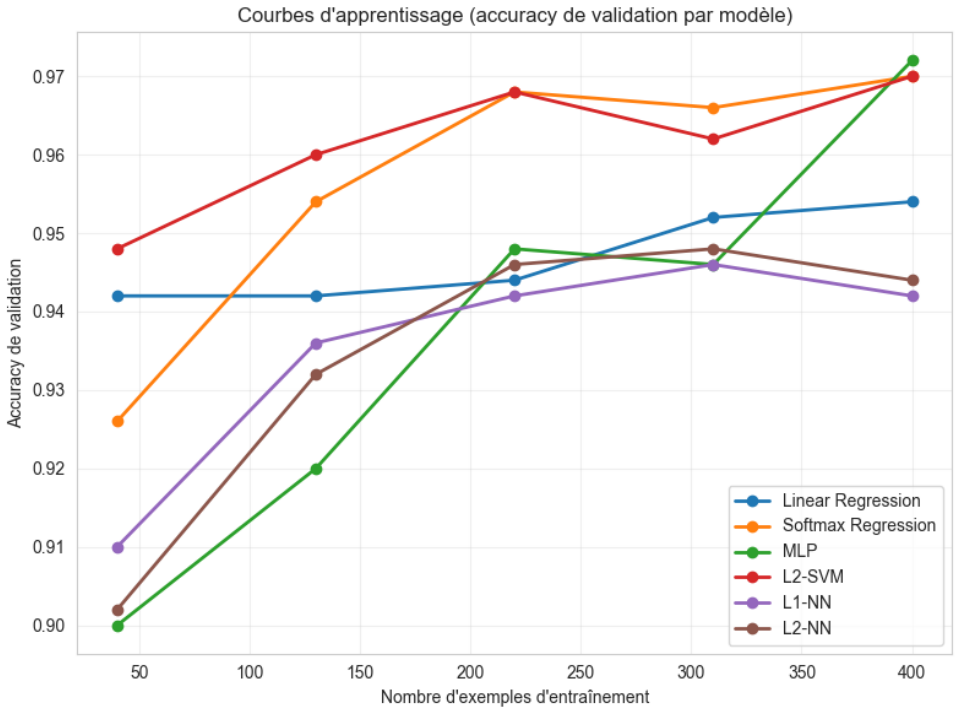


FIGURE 4 – Courbes d'apprentissage (accuracy train/validation vs taille d'échantillon).