République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Béjaïa
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de Fin d'Études

Présenté pour l'obtention du diplôme de Master en
Mathématiques Appliquées

Spécialité : **Sciences des Données et Aide à la Décision**

L'explicabilité des réseaux de neurones

Présenté par : **Mlle Maroua Hadjerioua**

Encadré par : **Dr. Lhadi Bouzidi**

Soutenu le **29/06/2025**, devant le jury composé de :

Président : M. Z. Aoudia
Examinateur : M. R. Sahli
Encadrant : Dr. Lhadi Bouzidi

Année Universitaire : 2024–2025

Remerciements

Je remercie Dieu pour m'avoir permis d'achever ce travail dans les meilleures conditions. Je remercie également ma famille pour son soutien constant tout au long de mon parcours.

Je tiens à exprimer ma sincère reconnaissance à Dr. Lhadi Bouzidi, mon encadrant, pour son accompagnement de qualité, sa disponibilité, et ses conseils éclairés qui ont largement contribué à la réalisation de ce mémoire.

Je souhaite également remercier l'ensemble du corps enseignant de l'Université Abderrahmane Mira de Béjaïa, dont la compétence et le dévouement ont enrichi ma formation sur les plans académique et humain.

Ma gratitude s'adresse aussi au personnel administratif et technique, pour leur appui et leur efficacité tout au long de mon cursus.

Enfin, je remercie toutes les personnes, collègues et ami(e)s, qui ont apporté leur aide ou leur soutien, de près ou de loin, durant cette période de travail.

Dédicace

À Allah, Le Miséricordieux,
source de toute lumière, de toute force et de toute paix.
C'est par Sa guidance que j'ai trouvé le chemin,
et par Sa miséricorde que j'ai transformé les blessures en volonté.

À la jeune fille de 15 ans que j'étais,
à celle qui, après la perte de sa mère, a pensé que tout s'arrêtait là.
Tu as voulu abandonner, et pourtant... tu as continué.
Tu as traversé la douleur, tu as affronté la solitude, tu as avancé pas à pas.
Aujourd'hui, tu as 23 ans.
Tu es devenue une femme forte, lucide, capable.
Tu sais ce que tu veux, et tu es prête à tout donner pour atteindre tes objectifs.
Et je veux que tu saches ceci : **je suis fière de toi**.

À ma mère,
ton amour ne m'a jamais quittée.
Même dans ton absence, tu as été une présence.
Ta voix, ta tendresse et ton souvenir m'ont guidée dans mes choix et mes silences.

À mon père,
pour sa patience, son soutien et sa confiance inébranlable.
Merci d'avoir été un refuge et une force,
et de m'avoir fait croire en l'amitié entre un père et sa fille.

À ma sœur, ma lumière,
à mes frères, mes piliers silencieux,
et à **mon amie précieuse**, fidèle et présente dans les moments clés.

À toutes les femmes qui tombent et se relèvent,
à celles qui doutent mais avancent quand même,
à celles qui construisent leur force dans le silence.

Et à mon peuple, le peuple de Palestine,
symbole de courage, d'espoir et de dignité inaltérable.

Résumé

Ce mémoire traite de l'explicabilité des réseaux de neurones appliqués à l'imagerie médicale. Face à la complexité croissante des modèles dits "boîtes noires", l'interprétation de leurs décisions devient un enjeu crucial, notamment dans les domaines sensibles comme la santé.

Dans une première partie, nous introduisons les fondements des réseaux de neurones profonds, en insistant sur leur architecture, leurs performances, mais aussi leurs limites. Une attention particulière est portée sur les risques d'overfitting, les stratégies de régularisation, et l'importance des métriques d'évaluation dans un contexte médical.

La deuxième partie est consacrée aux méthodes d'explicabilité. Nous distinguons les approches locales et globales, visuelles et non visuelles, en détaillant des techniques telles que Saliency Maps, Integrated Gradients, Grad-CAM, SHAP, LIME, ou encore les modèles substituts. Chaque méthode est replacée dans un cadre théorique rigoureux et illustrée à travers des cas concrets.

Enfin, une troisième partie propose une application pratique sur plusieurs jeux de données médicaux, notamment DermaMNIST, BreastMNIST, Breast Cancer Wisconsin, Pima Indians Diabetes et Cleveland Heart Disease. Nous y analysons l'apport réel des méthodes explicatives pour le praticien, en comparant les visualisations générées, leur lisibilité, et leur pertinence clinique.

Ce travail vise ainsi à concilier la performance des modèles de deep learning avec une exigence de transparence, essentielle dans le domaine médical.

Table des matières

Table des figures

Liste des tableaux

Introduction

Ces dernières années, l'intelligence artificielle, et en particulier le *deep learning*, a profondément transformé des domaines comme la reconnaissance d'images, la traduction automatique et l'aide au diagnostic médical. Si ces modèles atteignent des performances remarquables, leur opacité soulève une question cruciale, notamment en médecine : peut-on leur faire confiance sans comprendre leurs décisions ?

Dans un contexte médical, où chaque prédiction peut avoir un impact vital, l'explicabilité devient un enjeu central. Il ne suffit pas qu'un modèle fournisse une réponse exacte : il doit aussi être capable d'expliquer sa décision de manière claire, afin de favoriser la confiance, la transparence, et la collaboration homme-machine.

Ce mémoire s'inscrit dans cette problématique. Il vise à explorer les méthodes d'explicabilité appliquées aux réseaux de neurones dans le cadre de l'imagerie médicale et de données cliniques.

Le premier chapitre présente les bases des réseaux de neurones : architectures, apprentissage, performances, et limites, notamment en lien avec la confiance dans les prédictions.

Le deuxième chapitre introduit les principales techniques d'explicabilité, des approches locales (Saliency Maps, LIME, SHAP) aux approches globales, avec une attention particulière portée aux exigences du domaine médical.

Le troisième chapitre propose une application concrète sur plusieurs jeux de données médicaux. Les modèles sont analysés à travers des visualisations explicatives, en évaluant leur clarté, leur robustesse, et leur utilité pour les praticiens.

Ce travail ne prétend pas répondre à tous les défis de l'IA en médecine, mais il ambitionne de montrer qu'une IA performante peut aussi être facile à comprendre, si l'on accorde à l'explicabilité la place qu'elle mérite.

Chapitre 1

Les réseaux de neurones artificiels

L'intelligence artificielle (IA) cherche à reproduire certaines capacités humaines comme l'apprentissage ou le raisonnement. Le machine learning, l'une de ses branches principales, permet aux machines d'apprendre à partir de données sans instructions explicites.

Le deep learning, basé sur les réseaux de neurones artificiels, a connu un fort développement. Ces modèles, composés de couches interconnectées, apprennent à extraire automatiquement des informations à partir de données complexes. Ils sont aujourd'hui largement utilisés dans des domaines comme la vision par ordinateur, le langage ou l'imagerie médicale.

Inspirés du cerveau humain, les réseaux de neurones traitent les données via des connexions pondérées et des fonctions non linéaires. Leurs architectures varient selon les tâches, incluant des modèles comme les réseaux convolutifs (CNN) ou récurrents (RNN).

Toutefois, malgré leurs performances, ces modèles sont souvent perçus comme des « boîtes noires », difficiles à interpréter. Cela pose un défi majeur : comprendre et expliquer leurs décisions, notamment dans des domaines sensibles comme la santé.

Ce chapitre présente les bases des réseaux de neurones – leur fonctionnement, leurs architectures et leur apprentissage – afin de poser les fondations nécessaires à l'étude de leur explicabilité dans les parties suivantes.

1.1 Histoire et Évolution

L'évolution des réseaux de neurones artificiels s'est faite par étapes, avec plusieurs avancées marquantes au fil des décennies :

- **1943** : Warren McCulloch et Walter Pitts proposent le premier modèle théorique de neurone artificiel dans leur article *A Logical Calculus of the Ideas Immanent in Nervous Activity*. Ce neurone, modélisé comme une unité logique binaire, pose les fondations des réseaux de neurones modernes, bien qu'il ne permette pas l'apprentissage automatique [15, 32].
- **1958** : Frank Rosenblatt introduit le **perceptron**, un modèle capable d'apprendre à partir d'exemples pour résoudre des problèmes linéairement séparables [20, 31].
- **Années 1970** : Les limites du perceptron pour traiter des problèmes non linéaires entraînent un déclin de l'intérêt pour les réseaux de neurones, ralentissant les recherches dans le domaine.

Le domaine connaît un nouvel essor à partir des années 1980 grâce à des avancées théoriques et techniques majeures :

— **Années 1980** : Introduction de fonctions d'activation non linéaires telles que *tanh* et $ReLU$, et redécouverte de l'algorithme de **rétropropagation**, permettant l'entraînement efficace des réseaux multicouches [30].
— **Début des années 1990** :
  — **Yann LeCun** développe les **réseaux convolutifs** (CNN), adaptés à la reconnaissance d'images grâce à l'extraction automatique de caractéristiques spatiales [19].
  — En **1997**, les réseaux **LSTM** (Long Short-Term Memory) sont introduits pour traiter les données séquentielles, notamment en reconnaissance vocale et en traduction automatique.
— **2012** : Le réseau **AlexNet** remporte la compétition **ImageNet**, marquant une avancée décisive dans le domaine de la classification d'images par des réseaux profonds [4, 5].
— **Depuis 2017** : Les architectures à base d'**attention**, en particulier les **Transformers**, révolutionnent le **traitement du langage naturel** (NLP), en permettant des performances inédites en compréhension, génération et traduction de texte.
— **Depuis les années 2010** : Avec l'augmentation de la complexité des modèles, la question de leur **explicabilité** devient cruciale, notamment dans des domaines sensibles comme la santé ou la justice. Cela conduit au développement de méthodes d'analyse et d'interprétation des décisions des réseaux de neurones, regroupées sous le terme XAI (eXplainable Artificial Intelligence).

1.2 Architecture de base d'un réseau de neurone

1.2.1 Le Perceptron

Le perceptron représente l'une des premières tentatives de modélisation d'un neurone artificiel capable d'apprendre à partir de données [20]. Bien que simple, ce modèle a jeté les bases des algorithmes modernes d'apprentissage automatique [11]. C'est l'unité fondamentale des réseaux de neurones. Il s'agit d'un modèle de classification binaire capable de séparer linéairement deux classes de points.

Modèle Mathématique Le perceptron utilise un modèle linéaire pour représenter une frontière de décision. Formellement, étant donné une entrée $x = (x_1, x_2, \ldots, x_m)$, le perceptron retourne :

$$y = \begin{cases} +1 & \text{si } w_0 + \sum_{i=1}^{m} w_i x_i > 0, \\ -1 & \text{sinon.} \end{cases}$$

Cette équation définit un hyperplan affine qui sépare les deux classes [3].

Exemple : Classification de Plantes Toxiques Considérons un exemple concret de classification de plantes toxiques et non toxiques. Les données sont linéairement séparables, ce qui permet d'utiliser un perceptron pour tracer une frontière de décision (Fig. 1.1).

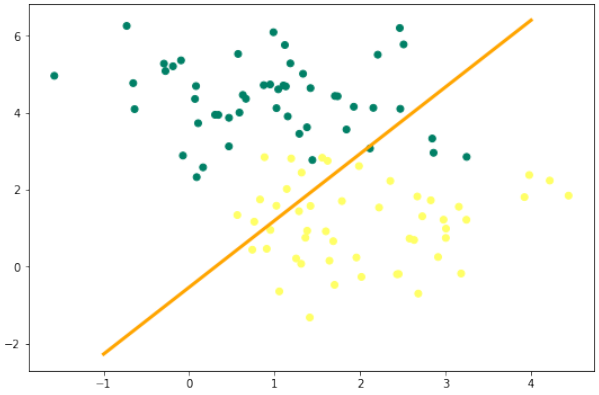

FIGURE 1.1 – Frontière de décision pour la classification de plantes (adapté de [12]).

Fonction d'Activation : Sigmoïde Pour améliorer le modèle, nous introduisons une fonction d'activation sigmoïde :

$$a(Z) = \frac{1}{1 + e^{-Z}}$$

Cette fonction convertit la sortie Z en une probabilité, permettant une classification plus nuancée [30].

Algorithme d'Apprentissage L'algorithme d'apprentissage du perceptron ajuste les poids w en fonction des erreurs commises sur une base d'exemples (x_n, y_n) [20] :

1. Initialiser les poids w à zéro
2. Pour chaque exemple (x_n, y_n) :
   — Calculer la prédiction $\hat{y}_n = \text{signe}(w \cdot x_n)$
   — Si $\hat{y}_n \neq y_n$, mettre à jour les poids : $w \leftarrow w + y_n x_n$
3. Répéter jusqu'à convergence [12]

Limitations du Perceptron Le perceptron ne peut résoudre que des problèmes linéairement séparables, comme le montre l'exemple du XOR (Fig. 1.2). Cette limitation fondamentale a été formalisée par [16].

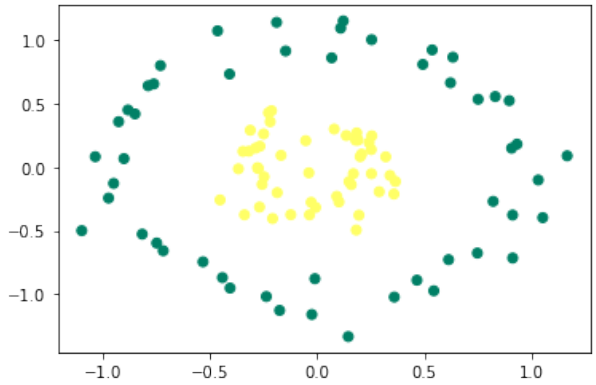

FIGURE 1.2 – Problème du XOR : non-linéairement séparable (source : [16]).

1.2.2 Fonctions d'activation

Les fonctions d'activation sont essentielles pour introduire de la non-linéarité dans les réseaux de neurones, leur permettant de modéliser des relations complexes dans les données [8, 14]. Le tableau ci-dessous résume les fonctions les plus courantes [8, 2] :

Fonction	Formule	Caractéristiques
Step	$f(z) = \begin{cases} 0 & \text{si } z < 0 \\ 1 & \text{si } z \geq 0 \end{cases}$	Utilisée pour la classification binaire, très simple mais non différentiable.
Sigmoïde	$f(z) = \frac{1}{1+e^{-z}}$	Produit une sortie entre 0 et 1 (probabilité). Problème de *vanishing gradient* pour z extrême [22].
ReLU	$f(z) = \max(0, z)$	Très utilisée en deep learning. Simple et efficace, évite souvent les gradients nuls [2, 8].
Softmax	$f(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}}$	Utilisée en sortie pour la classification multi-classes. Produit des probabilités normalisées.

TABLE 1.1 – Résumé des fonctions d'activation courantes.

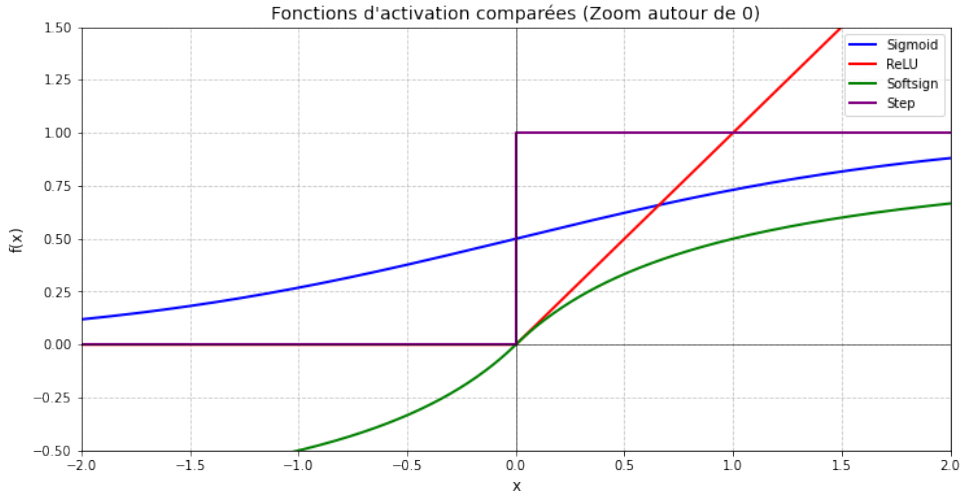

FIGURE 1.3 – Comparaison graphique des principales fonctions d'activation.

1.2.3 Architecture des Réseaux de Neurones ou Perceptron Multicouches

L'architecture d'un réseau de neurones définit la manière dont les neurones sont organisés et connectés. Elle est composée de plusieurs éléments clés [8, 6, 13, 21, 7] :

1. Les Couches (Layers)

Un réseau de neurones est composé de plusieurs couches interconnectées :

— **Couche d'entrée (Input Layer)** : Reçoit les données d'entrée (images, nombres, texte, etc.). Chaque neurone de cette couche représente une caractéristique des données [8, 6].

— **Couches cachées (Hidden Layers)** : Effectuent des transformations non linéaires en appliquant des poids et des fonctions d'activation. Plus il y a de couches, plus le réseau est profond [8, 21].

— **Couche de sortie (Output Layer)** : Produit la prédiction finale (classification, régression, etc.) avec un nombre de neurones correspondant aux classes ou valeurs à prédire [6, 7].

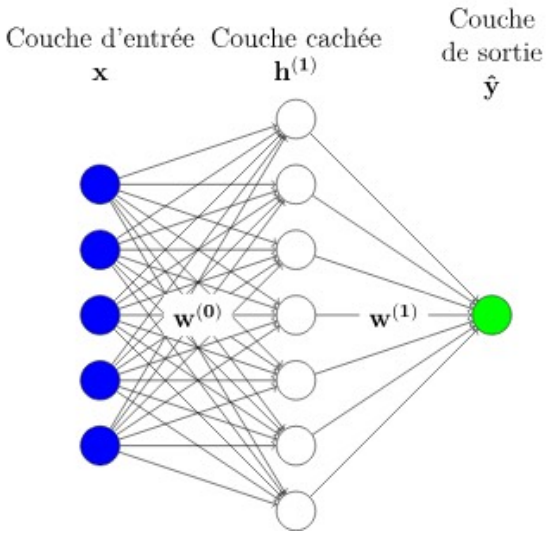

FIGURE 1.4 – Exemple d'architecture d'un perceptron multicouche.

2. Les Neurones (Neurons)

Chaque neurone reçoit les entrées pondérées, applique une fonction d'activation, puis transmet le résultat à la couche suivante [21] :

$$a_j^{(l)} = f\left(\sum_{i=1}^{n} w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$$

3. Les Connexions et Poids

Chaque connexion entre neurones est associée à un poids qui ajuste l'influence de l'entrée sur la sortie. Un biais est ajouté pour améliorer la flexibilité du modèle [13]. On distingue au moins 3 types de connexions :

— **Fully Connected (Dense)** : Chaque neurone d'une couche est connecté à tous les neurones de la couche suivante [8].
— **Convolutionnelles** : Utilisées dans les réseaux convolutifs (CNN) pour capturer des motifs locaux dans les images [8].
— **Récursive** : Utilisées dans les réseaux récurrents (RNN) pour traiter des séquences [6].

4. La fonction d'activation

Elle introduit la non-linéarité nécessaire pour que le réseau puisse apprendre des relations complexes [8].

1.2.4 Propagation Avant (Forward Propagation)

La **propagation avant** est une étape essentielle dans le fonctionnement d'un réseau de neurones. Elle consiste à propager les données d'entrée à travers les différentes couches du réseau pour produire une prédiction finale. Cette étape calcule les activations intermédiaires et finales en utilisant les poids (W) et biais (b) actuels [11, 12, 18].

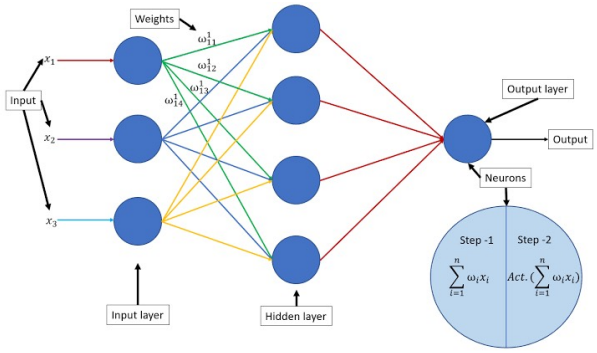

FIGURE 1.5 – Schéma de la propagation avant dans un réseau de neurones.

Étapes de la Propagation Avant

Pour chaque couche l, les étapes suivantes sont effectuées :

1. **Calcul des valeurs linéaires** ($Z^{[l]}$) :

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]},$$

où :
— $W^{[l]} \in \mathbb{R}^{n^{[l]} \times n^{[l-1]}}$: matrice des poids de la couche l,
— $A^{[l-1]} \in \mathbb{R}^{n^{[l-1]} \times m}$: activations de la couche précédente (ou les entrées si $l = 1$),
— $b^{[l]} \in \mathbb{R}^{n^{[l]} \times 1}$: vecteur des biais de la couche l,
— $Z^{[l]} \in \mathbb{R}^{n^{[l]} \times m}$: valeurs linéaires calculées pour la couche l.

2. **Application de la fonction d'activation** ($A^{[l]}$) :

$$A^{[l]} = g(Z^{[l]}),$$

où $g(\cdot)$ est la fonction d'activation (sigmoïde, ReLU, tanh, etc.) [11].

Ces étapes sont répétées pour chaque couche jusqu'à atteindre la couche de sortie.

Importance de la Propagation Avant

La propagation avant est cruciale car elle permet de :
— Calculer les prédictions finales du modèle,
— Introduire de la non-linéarité grâce aux fonctions d'activation,
— Préparer les résultats nécessaires pour la rétropropagation [11, 18].

1.2.5 Rétropropagation (Back Propagation)

La rétropropagation est une étape cruciale dans l'entraînement des réseaux de neurones. Elle permet de calculer les gradients des paramètres du modèle (poids W et biais b) par rapport à la fonction de coût, afin de les ajuster pour minimiser cette fonction [11, 12, 18]. Cette méthode repose sur deux principes fondamentaux :

1. **Règle de la Chaîne (Chain Rule)** : Permet de propager les erreurs depuis la couche de sortie vers les couches précédentes en calculant les dérivées partielles des paramètres.
2. **Descente de Gradient (Gradient Descent)** : Utilisée pour mettre à jour les poids et biais en suivant la direction opposée au gradient afin de minimiser la fonction de coût [11].

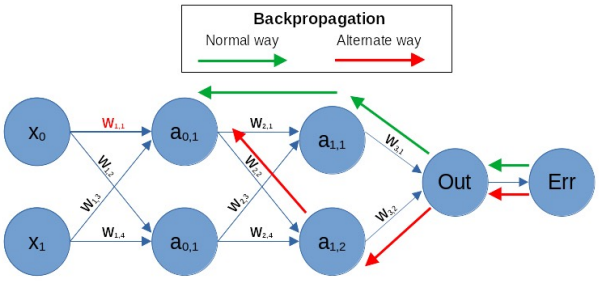

FIGURE 1.6 – Illustration du principe de rétropropagation.

1.3 Réseaux de neurones convolutif

Dans cette section, nous allons explorer l'un des algorithmes les plus efficaces du Deep Learning : les réseaux de neurones convolutifs (CNN). Ces modèles sont particulièrement puissants pour traiter les données visuelles, notamment en ce qui concerne la reconnaissance d'images. Les CNN sont capables d'attribuer automatiquement une étiquette à chaque image en fonction de sa classe d'appartenance, grâce à leur capacité à extraire des caractéristiques pertinentes directement à partir des données d'entrée. Cela les rend essentiels pour diverses applications en intelligence artificielle, notamment la vision par ordinateur et le traitement d'images.

Reconnaissance d'Images et Vidéos

Les CNN sont particulièrement efficaces pour la classification et la reconnaissance d'images, offrant un apprentissage rapide et un faible taux d'erreur. Ils sont également appliqués à l'analyse vidéo .

Traitement du Langage Naturel

Bien que moins courant, les CNN peuvent être utilisés pour l'analyse sémantique, la modélisation de phrases, la classification et la traduction. Ils offrent une représentation contextuelle du langage sans hypothèse de séquence . **Découverte de Médicaments** Les CNN sont utilisés pour prédire les interactions entre molécules et protéines biologiques, aidant ainsi à identifier des traitements potentiels .[9]

Jeux et Intelligence Artificielle

Ils ont été utilisés avec succès dans des logiciels de jeu comme le Go et les échecs. De plus, les CNN sont capables de détecter les anomalies dans les images .

Autres Applications

Les CNN sont également utilisés dans des systèmes de recommandation et pour la reconnaissance vocale

1.3.1 Architecture d'un Convolutional Neural Network-CNN

Les Convolutional Neural Networks (CNN) sont une sous-catégorie de réseaux de neurones profonds, réputés pour leur excellence dans la classification d'images . Leur fonctionnement semble simple à première vue : l'utilisateur fournit une image sous forme de matrice de pixels. Cette matrice possède trois dimensions : deux pour les images en niveaux de gris et une troisième de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).[9]

Contrairement aux modèles MLP classiques, qui ne comportent qu'une partie de classification, l'architecture des CNN se divise en deux parties distinctes : une partie convolutive et une partie de classification

Partie Convolutive La partie convolutive vise à extraire des caractéristiques spécifiques à chaque image en les compressant pour réduire leur taille initiale. L'image passe à travers une succession de filtres, créant ainsi de nouvelles images appelées cartes de convolutions. Ces cartes sont ensuite concaténées en un vecteur de caractéristiques appelé code CNN

Partie Classification
Le code CNN obtenu en sortie de la partie convolutive est utilisé comme entrée pour une deuxième partie composée de couches entièrement connectées, appelées perceptron multicouche (MLP). Le rôle de cette partie est de combiner les caractéristiques du code CNN pour classer l'image

Schéma Représentant l'Architecture d'un CNN

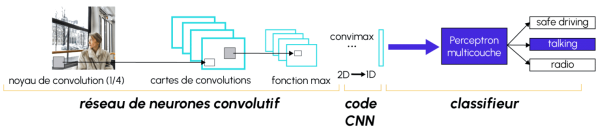

FIGURE 1.7 – Schéma Représentant l'Architecture d'un CNN

La partie convolutive joue un rôle crucial dans l'architecture des CNN. Mais à quoi sert la convolution ?

Convolution

La convolution est une opération mathématique utilisée pour le traitement et la reconnaissance d'images. Elle s'apparente à un filtrage. Sur une image, le processus commence par définir la taille de la fenêtre de filtre, qui se déplace progressivement sur l'image. À chaque portion d'image rencontrée, un calcul de convolution est effectué, produisant une carte d'activation ou feature map. Cette carte indique où les caractéristiques sont localisées dans l'image .[9]

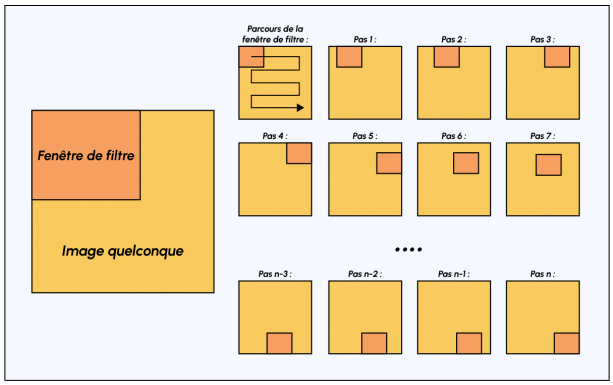

FIGURE 1.8 – Schéma du parcours de la fenêtre de filtre sur l'image

Exemple de Filtres de Convolution

Pendant la partie convolutive d'un CNN, l'image passe à travers plusieurs filtres de convolution. Parmi les filtres couramment utilisés, on trouve ceux permettant de détecter les bords ou les formes géométriques. Le choix et l'application de ces filtres se font automatiquement par le modèle. Des filtres comme le filtre moyenneur ou gaussien sont également utilisés pour réduire le bruit dans les images .

Effets des Filtres Moyenneur et Gaussien

Lorsqu'on applique ces filtres à une image bruyante, comme une photographie prise avec une faible luminosité, le filtre gaussien réduit le bruit sans altérer significativement la netteté, contrairement au filtre moyenneur

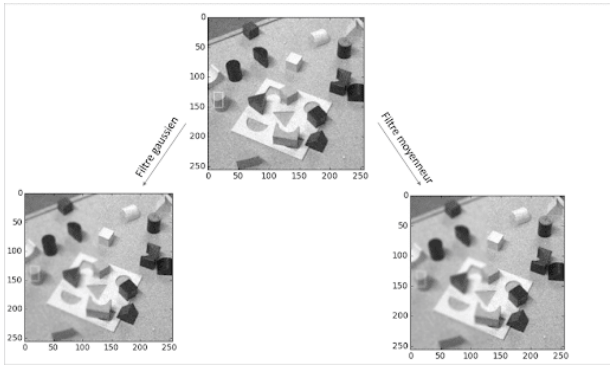

FIGURE 1.9 – Effet des filtres moyenneur et gaussien

Intérêt de la Partie Convolutive

Outre sa fonction de filtrage, la partie convolutive permet d'extraire des caractéristiques propres à chaque image en les compressant via des méthodes de sous-échantillonnage comme le Max-Pooling. Cela réduit la taille initiale des données tout en conservant les informations essentielles .[9]

1.3.2 Méthode de sous-échantillonnage : le Max-Pooling

Principe du Max-Pooling Le Max-Pooling est une technique de discrétisation basée sur des échantillons, utilisée pour réduire la dimension d'une représentation d'entrée, qu'il s'agisse d'une image ou d'une matrice de sortie d'une couche cachée . Cette méthode a pour avantage de diminuer le coût de calcul en réduisant le nombre de paramètres à apprendre. De plus, elle offre une certaine invariance aux petites translations, car si une petite translation ne modifie pas le maximum d'une région balayée, le maximum de chaque région restera inchangé, et donc la nouvelle matrice créée restera identique.[9]

Exemple concret

Imaginons une matrice 4×4 représentant notre entrée initiale et un filtre d'une fenêtre de taille 2×2 appliqué sur cette entrée. Pour chaque région balayée par le filtre, le Max-Pooling sélectionne la valeur maximale, créant ainsi une nouvelle matrice de sortie où chaque élément correspond aux maximums de chaque région rencontrée.

Processus de Max-Pooling

La fenêtre de filtre se déplace généralement de deux pixels vers la droite (stride/pas = 2) et récupère à chaque pas la valeur maximale parmi les valeurs de pixels de la région balayée

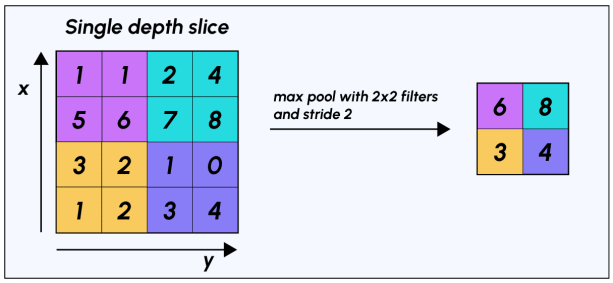

FIGURE 1.10 – Processus de Max-Pooling

Utilité du Max-Pooling dans les CNN L'ajout d'une partie convolutive en amont d'un modèle permet d'obtenir une "carte de caractéristiques" dont les dimensions sont plus petites que celles de l'image initiale. Cela réduit considérablement le nombre de paramètres à calculer dans le modèle, ce qui est un avantage majeur par rapport à un modèle MLP classique.[9]

Architecture d'un Convolutional Neural Network (CNN)

Une architecture typique d'un CNN se compose de plusieurs couches :

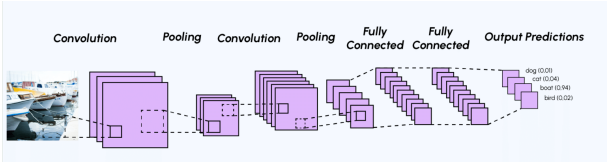

FIGURE 1.11 – Architecture d'un CNN

Couche de convolution (CONV)

Cette couche analyse les images fournies en entrée et détecte la présence de diverses caractéristiques. En sortie, on obtient un ensemble de cartes de caractéristiques

Couche de Pooling (POOL)

Généralement appliquée après une couche de convolution, cette couche réduit la taille des images tout en préservant leurs caractéristiques essentielles. Le Max-Pooling et l'Average Pooling sont des méthodes courantes utilisées ici. En sortie, on obtient le même nombre de cartes de caractéristiques mais avec une taille réduite .[9]

Couche d'activation ReLU (Rectified Linear Units)

Cette couche remplace toutes les valeurs négatives par des zéros, rendant le modèle non linéaire et plus complexecite

Couche Fully Connected (FC)

Placée en fin d'architecture, cette couche est entièrement connectée à tous les neurones de sortie. Elle applique une combinaison linéaire suivie d'une fonction d'activation pour classifier l'image d'entrée. En sortie, elle renvoie un vecteur de probabilités pour chaque classe.

Cette architecture permet aux CNN de traiter efficacement les données d'images tout en réduisant la complexité du modèle.[9]

1.4 Réseaux de neurones convolutifs (CNN)

Les CNN (Convolutional Neural Networks) sont parmi les modèles les plus performants pour traiter les données visuelles. Ils permettent l'extraction automatique de caractéristiques d'images et sont largement utilisés dans des domaines tels que la vision par ordinateur, la reconnaissance vocale, le traitement du langage naturel ou encore la découverte de médicaments [9]. Leur puissance les rend très attractifs dans le domaine médical, notamment pour l'analyse d'images médicales (radiographies, IRM, scanners), où ils peuvent détecter des anomalies avec une grande précision. Cependant, cette complexité soulève un défi majeur : comprendre et expliquer les décisions prises par ces modèles, en particulier dans des contextes sensibles comme la santé.

Applications courantes

— **Vision par ordinateur** : classification d'images et analyse vidéo avec un faible taux d'erreur.

— **NLP** : classification et modélisation de phrases, avec une approche non séquencée.
— **Bioinformatique** : prédiction d'interactions moléculaires.
— **Jeux** : détection d'anomalies et agents intelligents.
— **Autres** : systèmes de recommandation, reconnaissance vocale.

1.4.1 Architecture d'un CNN

Contrairement aux MLP, les CNN comportent deux parties :
— **Partie convolutive** : extrait des caractéristiques via des filtres et produit des cartes appelées *feature maps*.
— **Partie classification** : un MLP traite le vecteur obtenu pour générer la prédiction.

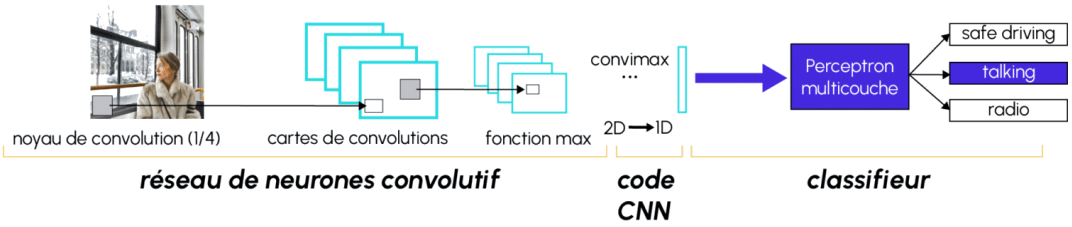

FIGURE 1.12 – Schéma de l'architecture d'un CNN.

1.4.2 Convolution et Filtres

La **convolution** est une opération qui fait glisser une fenêtre (filtre) sur l'image pour en extraire des motifs. Chaque filtre produit une *feature map* localisant des caractéristiques [9].

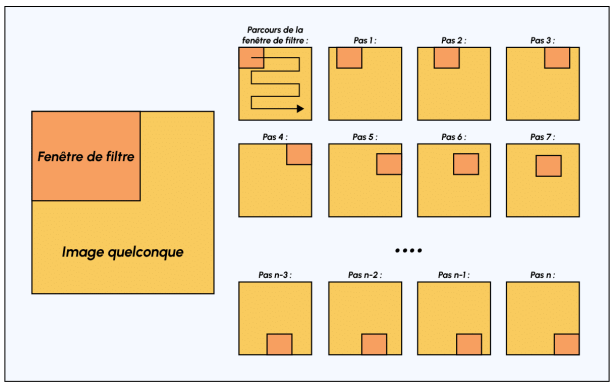

FIGURE 1.13 – Fenêtre de filtre glissant sur l'image.

Filtres usuels :
— **Bords / contours** : détection de formes.
— **Moyenneur / gaussien** : réduction du bruit.

1.4.3 Max-Pooling : sous-échantillonnage

Principe : le Max-Pooling réduit la taille des cartes de caractéristiques en ne conservant que le maximum local [9]. Cela réduit le coût de calcul tout en conservant les infor-

mations essentielles.

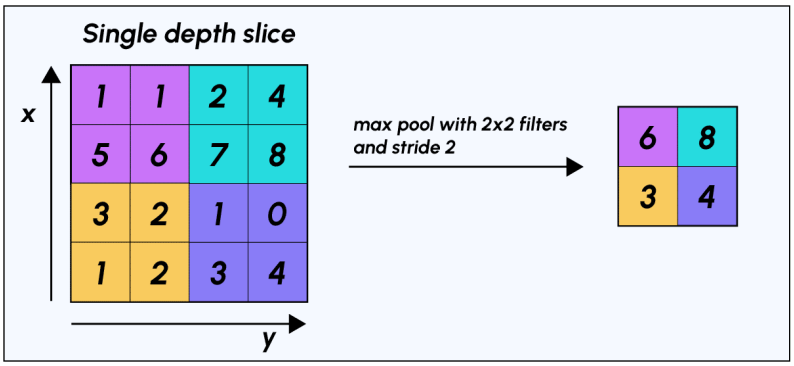

FIGURE 1.14 – Processus de Max-Pooling.

1.4.4 Structure typique d'un CNN

- **Convolution (CONV)** : extrait les caractéristiques de l'image.
- **Pooling (POOL)** : réduit les dimensions.
- **Activation (ReLU)** : introduit la non-linéarité.
- **Fully Connected (FC)** : effectue la classification finale [9].

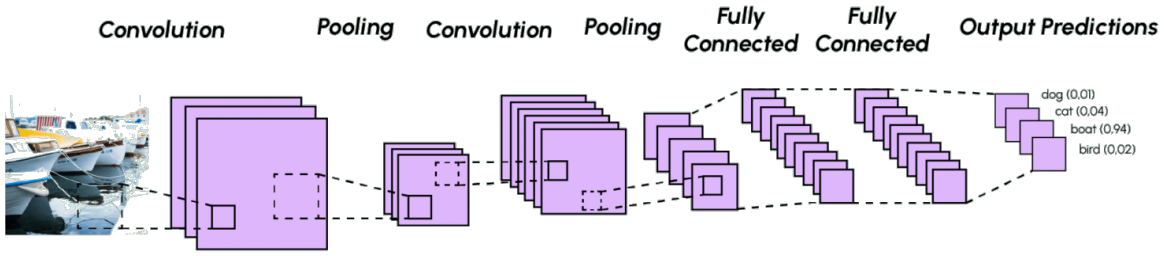

FIGURE 1.15 – Architecture typique d'un CNN.

Lien avec l'explicabilité

Malgré leur efficacité, les CNN sont souvent considérés comme des "boîtes noires". Dans un contexte médical, où les décisions doivent être justifiées et interprétables, il est crucial de développer des techniques permettant d'expliquer les prédictions d'un CNN (ex. : cartes de saillance, Grad-CAM, LRP). Ces méthodes visent à améliorer la transparence, la confiance et la sécurité des systèmes basés sur l'IA en santé.

1.5 Entraînement des réseaux de neurones

1.5.1 Fonctions de perte

Les fonctions de perte, ou fonctions de coût, quantifient l'écart entre les prédictions du modèle et les valeurs attendues. Elles guident l'ajustement des poids du réseau pendant l'apprentissage. Dans le contexte médical, bien choisir une fonction de perte est crucial pour minimiser les erreurs critiques. Deux fonctions courantes sont :

1. **Erreur quadratique moyenne (MSE)** [9] :

$$L(\mathcal{D}; Z) = \frac{1}{n} \sum_{i=1}^{n} \|\hat{y}_i - y_i\|^2$$

Amplifie les grandes erreurs, adaptée à la régression.

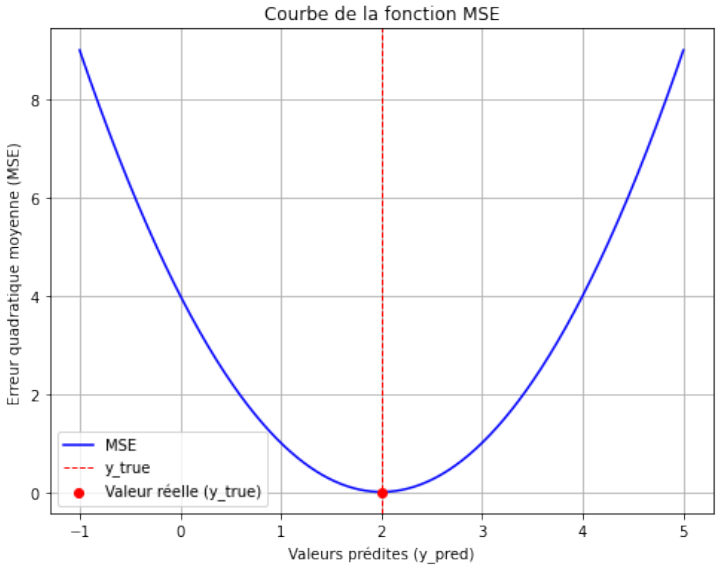

FIGURE 1.16 – Courbe de la fonction perte (MSE).

2. **Perte logistique (log loss)** :

$$L(\mathcal{D}; Z_\theta) = -\frac{1}{n} \sum_{i} \log p(\hat{y}_i = y_i; Z_\theta)$$

Appropriée pour la classification, elle pousse le modèle à donner une probabilité proche de 1 à la classe correcte.

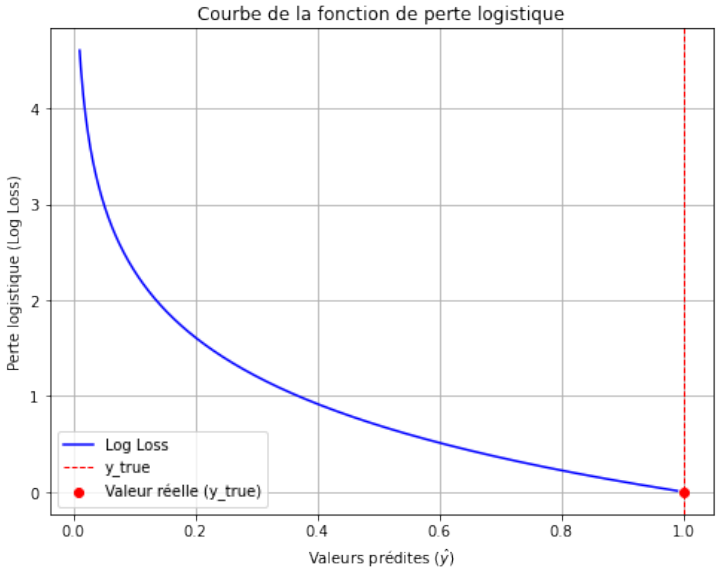

FIGURE 1.17 – Fonction de perte logistique.

1.5.2 Optimisation des paramètres

Pour minimiser la fonction de perte, on ajuste les poids du réseau par optimisation. Deux algorithmes sont largement utilisés :

Descente de gradient stochastique (SGD)

— **Mise à jour** :

$$W_{t+1} = W_t - \alpha \cdot \nabla_W L(W_t)$$

— **Avantages** : mise à jour fréquente, exploration de l'espace des paramètres.
— **Limites** : sensible au taux d'apprentissage, risque d'oscillations.

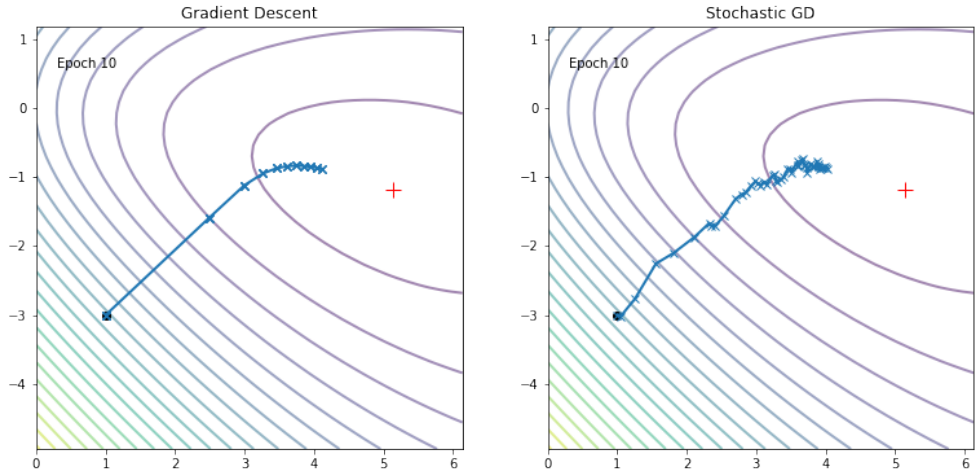

FIGURE 1.18 – Descente de gradient vs descente stochastique.

Adam (Adaptive Moment Estimation)

— Combine les méthodes Momentum et RMSProp.
— Utilise deux moments :

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_W L(W_t)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_W L(W_t))^2$$

— Correction de biais :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

— Mise à jour finale :

$$W_{t+1} = W_t - \alpha \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

— **Avantages** : taux adaptatif, convergence rapide.
— **Limites** : plus coûteux, nécessite un bon réglage des hyperparamètres.

1.5.3 Surapprentissage et sous-apprentissage

— **Underfitting (sous-apprentissage)** : modèle trop simple, mauvaise performance sur tous les jeux de données.
— **Overfitting (surapprentissage)** : modèle trop complexe, excellente performance sur l'entraînement mais mauvaise généralisation.

Observation : les courbes d'apprentissage aident à détecter ces problèmes (courbe de perte entraînement vs validation).

1.5.4 Stratégies de régularisation

Pour réduire l'overfitting, plusieurs techniques sont employées :
— **Dropout** : désactivation aléatoire de neurones à chaque itération.
— **Batch Normalization** : normalisation des activations pour stabiliser et accélérer l'entraînement.
— **Data Augmentation** : génération d'exemples synthétiques à partir des données existantes (rotations, zooms, etc.).

Ces stratégies sont essentielles dans le domaine médical, où les données sont souvent rares, et l'interprétabilité critique pour garantir des décisions fiables et compréhensibles.

1.6 Évaluation des modèles

La qualité d'un modèle de classification ne se limite pas à sa précision. D'autres métriques permettent une évaluation plus fine, notamment dans le contexte médical où les erreurs peuvent avoir des conséquences graves.

— **Précision (Accuracy)** : proportion d'échantillons correctement classés.
— **Rappel (Recall)** : proportion de vrais positifs détectés parmi les cas positifs réels.
— **F1-score** : moyenne harmonique entre précision et rappel, utile en cas de classes déséquilibrées.
— **Courbe ROC et AUC** : la courbe ROC trace le taux de vrais positifs contre le taux de faux positifs ; l'aire sous la courbe (AUC) mesure la capacité du modèle à distinguer les classes.

Exemple d'application : dans un modèle de classification binaire visant à détecter un cancer du sein à partir d'images ou de mesures cliniques, le rappel est souvent prioritaire. En effet, un faux négatif pourrait retarder la prise en charge d'un patient atteint, ce qui est inacceptable en pratique médicale. L'interprétation de ces métriques permet donc d'aligner l'évaluation du modèle avec les objectifs cliniques réels.

1.7 Limites des réseaux de neurones

Malgré leurs performances impressionnantes, les réseaux de neurones présentent certaines faiblesses majeures, particulièrement critiques dans les domaines sensibles comme la médecine :

— **Manque d'interprétabilité** : les décisions prises par un réseau de neurones sont souvent perçues comme issues d'une "boîte noire", rendant difficile la compréhension ou la justification d'une prédiction, en particulier dans les contextes médicaux où la transparence est essentielle.

— **Forte dépendance aux données** : ces modèles nécessitent de grandes quantités de données annotées pour bien fonctionner. Des données biaisées, bruitées ou insuffisantes peuvent compromettre gravement leur fiabilité.
— **Vulnérabilité aux attaques adversariales** : de légères perturbations imperceptibles peuvent suffire à tromper un modèle, le poussant à produire des résultats erronés.

Dans des domaines où les décisions ont des conséquences critiques, comme la santé, ces limites soulèvent une question fondamentale : *comment faire confiance à un modèle que l'on ne comprend pas* ?

C'est dans cette perspective que s'inscrit le chapitre suivant, consacré à l'explicabilité des réseaux de neurones. Il explore les approches permettant de rendre plus transparentes les prédictions des modèles, de vérifier leur cohérence avec les connaissances médicales, et de renforcer la confiance des utilisateurs humains dans les systèmes d'IA.

Chapitre 2

L'explicabilité des réseaux de neurones

Ce chapitre s'appuie principalement sur l'ouvrage de référence Interpretable Machine Learning de Christoph Molnar [17], dans lequel l'auteur présente les principaux concepts et méthodes liés à l'explicabilité des modèles d'apprentissage automatique. Les différentes approches décrites seront adaptées à notre contexte d'étude, qui porte à la fois sur des données visuelles (images médicales) et sur des données non visuelles (données cliniques tabulaires). L'objectif est de comprendre comment les modèles exploitent ces différentes sources d'information pour produire leurs prédictions, et d'identifier les facteurs qui influencent le plus les décisions prises.

2.1 Introduction

Le chapitre précédent a présenté les fondements des réseaux de neurones, leur architecture, leurs mécanismes d'apprentissage et leurs applications dans divers domaines, notamment en médecine. Leur aptitude à modéliser des relations complexes en fait des outils puissants pour la reconnaissance d'images ou encore l'interprétation de données cliniques. Toutefois, cette efficacité s'accompagne d'une limitation majeure : leur manque de transparence.

Souvent considérés comme des **boîtes noires**, les réseaux neuronaux profonds rendent difficile la compréhension des raisons d'une prédiction donnée. Dans des domaines sensibles comme la santé, où les décisions peuvent avoir des conséquences directes sur les patients, cette opacité pose un véritable problème. Comprendre pourquoi un modèle a pris une décision est essentiel pour gagner la confiance des professionnels de santé et répondre aux exigences éthiques et réglementaires.

Comme le souligne **Molnar** [17], l'explicabilité ne se limite pas à une meilleure compréhension par les experts en intelligence artificielle. Elle vise à fournir des justifications claires et accessibles à tous les utilisateurs finaux, qu'ils soient médecins, patients ou décideurs.

Face à ce besoin croissant de transparence, de nombreuses approches d'explicabilité ont émergé. Certaines cherchent à expliquer des prédictions individuelles (approches locales), d'autres à comprendre le comportement global du modèle (approches globales). Le présent chapitre se propose d'explorer ces méthodes, en détaillant leurs principes, avantages et limites, ainsi que leur applicabilité concrète aux données médicales, qu'elles soient visuelles ou textuelles.

2.2 Définition et enjeux de l'explicabilité

L'explicabilité, dans le domaine de l'intelligence artificielle, fait référence à la capacité d'un modèle à justifier ou rendre compréhensibles ses décisions. Elle se distingue de l'interprétabilité, qui désigne plutôt la faculté d'un humain à appréhender le fonctionnement global d'un modèle. L'explicabilité se concentre sur la compréhension de prédictions spécifiques, ce qui est crucial avec des modèles complexes comme les réseaux neuronaux profonds.

Souvent qualifiés de **boîtes noires**, ces modèles ne permettent pas d'identifier intuitivement les facteurs expliquant une décision. Leur structure complexe, faite de couches et de millions de paramètres, ne fournit pas spontanément d'indications claires sur les variables déterminantes, contrairement à des modèles plus simples comme les arbres de décision.

Dans des domaines sensibles tels que la **santé**, cette opacité devient problématique. Un professionnel de santé ne peut raisonnablement fonder une décision clinique sur la seule sortie d'un algorithme, sans comprendre ses fondements. Cela pose des enjeux de confiance, de responsabilité, mais aussi de conformité aux normes **réglementaires** telles que le RGPD, qui impose des exigences de transparence, notamment lorsqu'il s'agit de données personnelles ou de diagnostics.

Face à ces enjeux, un champ de recherche actif s'est développé autour de l'explicabilité des modèles d'apprentissage. L'objectif n'est pas de sacrifier la performance au profit de la transparence, mais de trouver un équilibre entre les deux. Ainsi, de nombreuses méthodes ont été proposées, qu'elles soient **locales** (centrées sur une prédiction) ou **globales** (centrées sur le comportement du modèle dans son ensemble). Ce chapitre présente ces approches et leurs applications dans le contexte médical.

2.2.1 Approches locales d'explicabilité

Parmi les différentes approches conçues pour interpréter les modèles complexes, les méthodes locales occupent une place importante. Elles permettent d'expliquer une prédiction particulière en identifiant les caractéristiques de l'entrée ayant le plus influencé la décision du modèle. Cette granularité est essentielle dans le domaine médical, où chaque cas individuel peut avoir un impact concret sur la prise en charge d'un patient.

Nous présentons ci-dessous trois grandes familles de méthodes locales, fondées sur des principes distincts :

- **LIME** : s'appuie sur une approximation locale du modèle pour simuler son comportement autour d'une prédiction cible ;
- **SHAP** : repose sur la théorie des jeux coopératifs pour quantifier la contribution de chaque caractéristique ;
- **Méthodes par gradients** : exploitent les dérivées du modèle, notamment en vision par ordinateur, pour identifier les régions les plus sensibles de l'entrée.

2.2.2 LIME (Local Interpertable Model-agnostic Explainations)

LIME est une méthode proposée en 2016 par Ribeiro, Singh et Guestrin. Elle vise à expliquer les prédictions individuelles des modèles d'apprentissage automatique dits « boîtes noires », comme les réseaux de neurones. L'idée principale est de comprendre pourquoi un modèle a produit une certaine prédiction, en construisant une approximation locale interprétable de ce modèle complexe.

Concrètement, LIME génère des variations artificielles de l'instance à expliquer (en introduisant de petites perturbations dans les données d'entrée), puis observe comment ces modifications influencent les prédictions du modèle. Cela permet de capturer le comportement local du modèle autour de cette instance.

Sur la base de ces données perturbées et des prédictions correspondantes du modèle boîte noire, LIME entraîne un modèle interprétable local (par exemple, une régression linéaire). Ce modèle simplifié doit approximer au mieux le comportement du modèle original, mais uniquement dans le voisinage de l'instance considérée.

Le processus peut être formalisé par l'optimisation suivante :

$$\text{Explanation}(x) = \arg\min_{g \in G}; L(f, g, \pi_x) + \Omega(g) \tag{2.1}$$

où :

— f est le modèle boîte noire,
— g est un modèle interprétable appartenant à une classe de modèles simples G,
— $L(f, g, \pi_x)$ mesure la fidélité locale entre g et f, pondérée par une mesure de proximité π_x,
— $\Omega(g)$ pénalise la complexité de g afin de garantir son interprétabilité.

Ainsi, LIME permet d'offrir une interprétation simple et compréhensible de décisions issues de modèles très complexes, en se concentrant uniquement sur ce qui se passe localement autour de chaque prédiction. [17]

Processus pratique de LIME

Dans la pratique, LIME n'optimise que la fonction de perte. La gestion de la complexité du modèle interprétable (comme le nombre de caractéristiques à conserver dans la régression linéaire) est laissée à la discrétion de l'utilisateur. Autrement dit, c'est à l'utilisateur de fixer, par exemple, le nombre maximal de variables explicatives que le modèle local est autorisé à utiliser, afin de garantir une interprétation claire.

La méthode suit plusieurs étapes pour construire ces modèles de substitution locaux :

1. **Sélection de l'instance à expliquer** : on commence par choisir un exemple spécifique (image, vecteur de données, etc.) pour lequel on souhaite comprendre la prédiction fournie par le modèle boîte noire.
2. **Perturbation de l'ensemble de données** : on génère un grand nombre de versions légèrement modifiées de cette instance (par exemple en activant ou désactivant des parties de l'image, ou en modifiant certaines valeurs des attributs).
3. **Prédictions sur les données perturbées** : chaque échantillon généré est ensuite évalué par le modèle complexe afin d'obtenir ses prédictions correspondantes.
4. **Pondération par proximité** : les nouveaux échantillons sont pondérés en fonction de leur ressemblance avec l'instance d'origine, à l'aide d'une fonction de similarité. Cela permet de privilégier les points les plus proches dans l'espace des caractéristiques.
5. **Entraînement du modèle local** : un modèle simple et interprétable (comme une régression linéaire) est alors entraîné sur cet ensemble pondéré. Ce modèle a pour objectif de reproduire localement les prédictions du modèle boîte noire, à partir des variations générées autour de l'instance cible.

[17]

Le modèle de substitution interprétable

Lorsqu'il cherche à expliquer une prédiction, LIME construit un petit modèle local, souvent une régression linéaire, dont le rôle est de reproduire localement la décision du modèle complexe (dans notre cas, un réseau de neurones). Ce modèle est volontairement simple et interprétable, de façon à ce qu'un humain puisse le comprendre sans difficulté. [17]

Sélection d'un nombre limité de caractéristiques

Afin que le modèle local reste compréhensible, LIME limite le nombre de variables explicatives utilisées (généralement entre 5 et 10). Cela évite une surcharge cognitive et met en évidence les caractéristiques les plus influentes. [17]

Utilisation de la régression Lasso

La régression Lasso est souvent utilisée dans ce cadre, car elle applique une pénalisation (L_1) sur les coefficients de la régression, ce qui pousse certains d'entre eux à devenir exactement nuls. Ainsi, les variables peu informatives sont automatiquement éliminées, simplifiant davantage le modèle.

L'impact du paramètre de régularisation est le suivant :

— Si la pénalisation est forte, le modèle conserve très peu de variables (voire aucune).
— En réduisant progressivement cette pénalisation, Lasso commence à inclure les variables jugées pertinentes, c'est-à-dire celles dont les coefficients deviennent non nuls.

Ce mécanisme permet à LIME de contrôler directement la taille du modèle explicatif. [17]

Stratégies de sélection des variables

Deux approches principales peuvent être utilisées pour sélectionner les variables explicatives du modèle de substitution :

— **Sélection en avant (forward selection)** : on part d'un modèle vide (contenant uniquement l'intercept) et on ajoute progressivement les variables, en choisissant à chaque étape celle qui améliore le plus la prédiction locale.
— **Sélection en arrière (backward selection)** : à l'inverse, on commence avec toutes les variables disponibles, puis on les retire une à une, en éliminant celles ayant le moins d'impact sur la prédiction, jusqu'à atteindre le nombre souhaité.

[17]

Génération des données locales

Un élément fondamental de LIME est qu'il ne dispose initialement que d'un seul exemple à expliquer. Il faut donc générer un ensemble de données locales autour de cette instance cible. Ces variations dépendent du type de données :

— **Données textuelles** : LIME modifie ou supprime certains mots du texte original.
— **Images** : LIME masque certaines zones appelées *superpixels* (groupes de pixels voisins similaires), afin d'évaluer leur influence sur la prédiction.

— **Données tabulaires** : LIME génère des échantillons en modifiant certaines valeurs numériques, en les remplaçant par des valeurs tirées d'une distribution normale centrée sur la moyenne, avec un écart-type estimé à partir de l'ensemble des données.

[17]

LIME pour les données tabulaires

Les données tabulaires correspondent à des tableaux structurés, similaires à ceux d'Excel : chaque ligne représente un exemple (par exemple un patient), et chaque colonne une variable ou caractéristique (par exemple l'âge, le poids, la tension artérielle, etc.). Lorsque

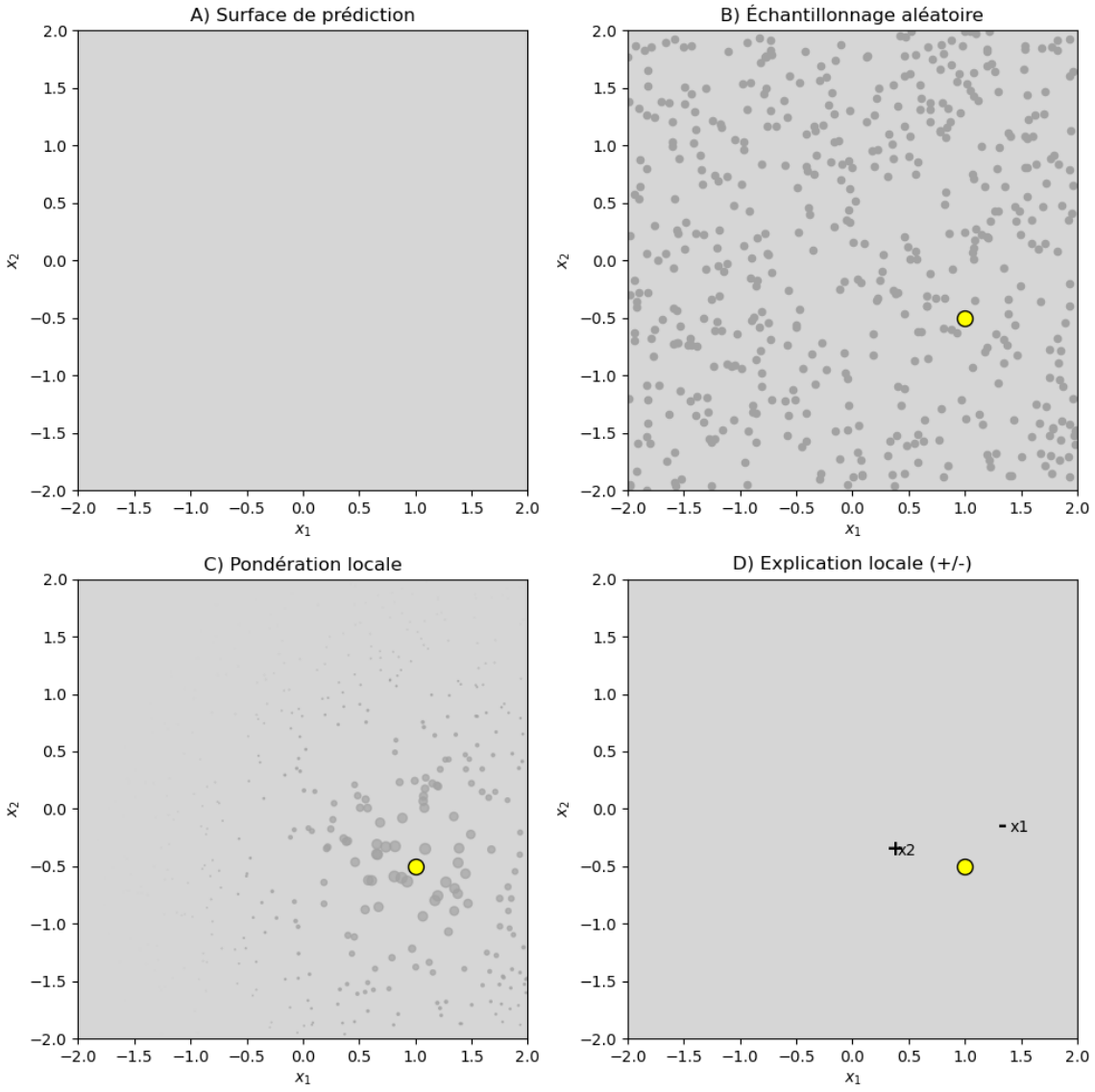

FIGURE 2.1 – Visualisation du mécanisme de LIME : échantillonnage local + apprentissage d'un modèle interprétable.

LIME crée des exemples artificiels autour de l'exemple que l'on souhaite expliquer, ces exemples ne sont pas toujours très proches de cet exemple d'origine. En effet, ils sont souvent générés à partir du centre des données, c'est-à-dire la moyenne de toutes les lignes du tableau.

Problème : cela signifie que ces exemples modifiés peuvent être très différents de celui que l'on essaie d'expliquer, et donc ne pas être toujours pertinents pour comprendre précisément ce cas particulier.

Toutefois, cette diversité des exemples peut aussi constituer un avantage. En effet, si les prédictions du modèle sur ces exemples générés diffèrent de la prédiction pour l'exemple de départ, LIME peut mieux comparer ces variations et ainsi mieux identifier les variables ayant réellement influencé la décision du modèle.

Ce qu'est un noyau de lissage dans LIME

LIME construit un modèle local autour de l'instance à expliquer. Pour cela, il doit déterminer quelles données environnantes (modifiées) doivent influencer ce modèle local.

Pour cela, LIME utilise une fonction appelée *noyau de lissage* (kernel), qui attribue un score de « proximité » entre un point généré et l'instance d'origine.

Ce score est élevé lorsque le point est proche (forte influence) et faible lorsqu'il est éloigné (faible influence).

Le noyau utilisé est un noyau exponentiel, qui calcule cette influence selon une formule dépendant d'un paramètre essentiel : la largeur du noyau.

La largeur du noyau (le cœur du problème)

Dans le code Python de LIME (`lime_tabular.py`), la largeur du noyau est fixée automatiquement à :

$$0.75 \times \sqrt{\text{nombre de colonnes}}$$

Ce choix est arbitraire, sans justification théorique rigoureuse.

Or, cette largeur a une influence majeure sur les résultats de l'explication :

— Si elle est trop petite, seules les données très proches influencent le modèle local, ce qui risque de provoquer un surapprentissage local (overfitting).

— Si elle est trop grande, des points éloignés mais moins pertinents influenceront aussi le modèle local, ce qui peut biaiser l'explication.

Conséquences pratiques

Il est possible d'inverser l'interprétation donnée par LIME simplement en modifiant la largeur du noyau.

De plus, LIME applique une distance (souvent euclidienne standard) entre les points dans l'espace des caractéristiques pour déterminer leur influence. Cela pose problème, car toutes les caractéristiques n'ont pas la même unité ni le même sens (par exemple, 1 cm de longueur de bec ne vaut pas 1 kg de poids).

Ainsi, selon la largeur du noyau choisie, la même caractéristique peut être jugée importante positivement, négativement, ou même non pertinente. Cette variabilité remet en question la fiabilité des explications fournies.

Exemple d'application médicale : prédiction du diabète avec un réseau de neurones

Considérons un réseau de neurones entraîné sur le jeu de données médical Pima Indians Diabetes, qui vise à prédire la présence ou l'absence de diabète de type 2 à partir de caractéristiques cliniques (taux de glucose, IMC, âge, etc.).

Pour comprendre la décision du modèle sur une patiente spécifique, LIME est utilisé afin de produire une explication locale, mettant en lumière les variables ayant le plus influencé la prédiction.

Exemple d'interprétation locale

Supposons qu'une patiente présente les caractéristiques suivantes :

Caractéristique	Valeur
Glucose	155
Pression artérielle	70
IMC	35.2
Âge	45
Nombre de grossesses	2

TABLE 2.1 – Caractéristiques cliniques de la patiente

Le réseau de neurones prédit un **risque élevé de diabète**.

En appliquant LIME, un ensemble de données artificielles est généré en perturbant légèrement ces variables. Chaque point modifié est évalué par le modèle, puis pondéré selon sa distance à l'instance originale via le noyau de lissage.

Rôle et impact de la largeur du noyau

LIME accorde plus de poids aux points proches de la patiente grâce au noyau exponentiel, mais la largeur du noyau modifie fortement le résultat :

— Une largeur faible donne plus d'importance au taux de glucose : de petites variations de cette variable modifient significativement la prédiction locale.

— Une largeur plus large fait apparaître l'IMC comme variable importante, et réduit l'impact relatif du glucose.

Ainsi, **les explications de LIME peuvent varier selon le paramètre de largeur du noyau**, ce qui soulève des questions sur la stabilité et la robustesse des interprétations dans un contexte clinique.[17]

LIME pour les données textuelles

Dans le cas des données textuelles, LIME adopte une stratégie différente de celle utilisée pour les données tabulaires. Plutôt que de modifier les valeurs de colonnes, la méthode génère des versions alternatives d'un texte en retirant certains mots de manière aléatoire.

Chaque nouvelle phrase ainsi obtenue est transformée en un vecteur binaire, où chaque position indique la présence (1) ou l'absence (0) d'un mot du texte original. Cette représentation permet de mesurer l'influence de chaque mot sur la prédiction du modèle, en apprenant un modèle local simple à partir de ces variantes textuelles. [17]

Étape 1 : Génération de textes voisins

Prenons comme point de départ une phrase tirée d'un rapport médical :

> *« Le patient présente une fièvre persistante accompagnée de douleurs thoraciques. »*

LIME crée des variantes de cette phrase en supprimant aléatoirement certains mots. Voici quelques exemples :

— Variante 1 : *« patient fièvre douleurs thoraciques »*

— Variante 2 : *« présente fièvre accompagnée thoraciques »*

— Variante 3 : *« Le patient douleurs thoraciques »*

Chaque variante est représentée sous forme d'un vecteur binaire, où chaque mot du texte initial est une colonne : 1 s'il est présent, 0 s'il est absent.

Le	patient	présente	fièvre	persistante	accompagnée	de	douleurs	thoraciques
1	1	1	1	0	1	0	1	1

TABLE 2.2 – Exemple de représentation binaire d'une variante

Étape 2 : Prédiction du modèle

Chaque texte modifié est soumis au modèle de prédiction (par exemple un réseau de neurones) qui évalue la probabilité d'un diagnostic donné, ici une *pneumonie probable*. On obtient des résultats comme :

Phrase	Probabilité de pneumonie	Poids (proximité)
patient fièvre douleurs thoraciques	0,81	0,76
présente fièvre accompagnée thoraciques	0,72	0,65
Le patient douleurs thoraciques	0,34	0,58

TABLE 2.3 – Prédiction du modèle et pondération des variantes

Étape 3 : Construction d'un modèle local

LIME ajuste ensuite un modèle simple, généralement une régression linéaire, sur les données voisines. Ce modèle local apprend à reproduire les sorties du modèle complexe à partir des vecteurs binaires (présence ou absence de mots).

Étape 4 : Interprétation

Les coefficients du modèle local indiquent l'importance de chaque mot dans la prédiction. Par exemple :

Cas	Probabilité	Mot	Poids
1	0,34	Le	0,00
1	0,34	douleurs	0,00
2	0,81	fièvre	3,92
2	0,81	thoraciques	2,15
2	0,81	persistante	0,00

TABLE 2.4 – Poids d'importance des mots selon le modèle local

Interprétation :

— **Cas 1** : La suppression de certains mots non significatifs n'impacte pas la prédiction, ce qui se traduit par des poids nuls.

— **Cas 2** : Les mots *« fièvre »* et *« thoraciques »* sont fortement corrélés avec la prédiction de pneumonie, ce qui leur confère des poids élevés.

LIME pour les données d'image

Contrairement aux données tabulaires, LIME applique des perturbations sur des *superpixels*, c'est-à-dire des groupes de pixels spatialement proches et de teinte similaire. Ces superpixels sont masqués ou modifiés afin de générer des variantes de l'image originale. À partir des prédictions du modèle sur ces variantes, LIME construit un modèle simple et local pour identifier les superpixels ayant le plus d'influence sur la prédiction [17].

Dans le cas illustré dans la figure ci-dessous, LIME a été appliqué à une image de chat tirée de la bibliothèque `skimage`, analysée par ResNet18, un réseau pré-entraîné sur

ImageNet. Ce modèle classe l'image comme « *tabby cat* » avec 64 % de probabilité et « *Egyptian cat* » avec 21 %. Cette proximité entre classes est typique dans ImageNet, qui inclut de nombreuses variantes de chats.

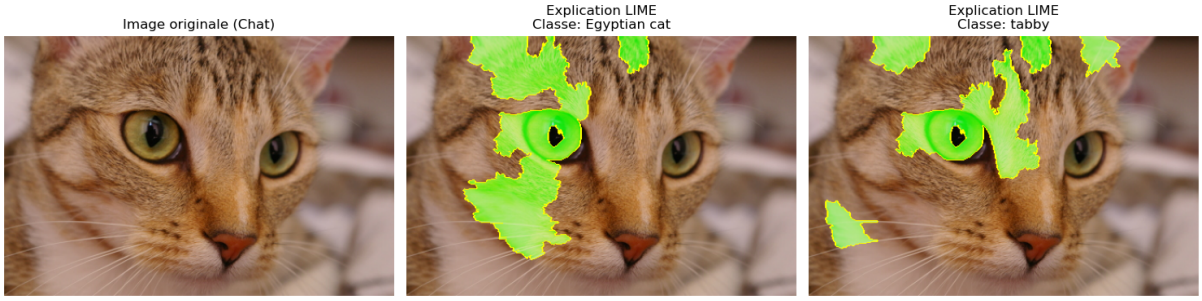

FIGURE 2.2 – Visualisation des explications LIME appliquées à une image de chat

Les résultats visuels montrent les zones influençant positivement (en vert) ou négativement (en rouge) la prédiction. Sur cette image, les parties du pelage et du visage du chat apparaissent majoritairement en vert, confirmant leur rôle clé. Très peu de zones rouges sont présentes, ce qui indique que le reste de l'image n'a pas d'impact négatif significatif.

Ainsi, LIME offre une interprétation intuitive des décisions d'un réseau convolutionnel, facilitant la validation du modèle et la détection d'éventuelles erreurs [17].

Points forts

LIME se distingue par sa flexibilité : qu'il s'agisse d'un SVM, d'un réseau de neurones ou de XGBoost, il permet d'obtenir des explications cohérentes et interprétables via un modèle simple (comme un arbre de décision), sans modifier le modèle d'origine.

Basé sur des méthodes éprouvées comme le Lasso ou les arbres peu profonds, LIME produit des explications concises, sélectives et contrastées, accessibles même aux non-experts. Il s'adapte également à différents types de données — textuelles, tabulaires ou images — ce qui en fait une méthode polyvalente.

En outre, LIME inclut une mesure de fidélité permettant d'évaluer la qualité locale de l'explication. Des bibliothèques disponibles en Python (lime) et en R (lime, iml) facilitent son utilisation dans des pipelines de machine learning.

Un atout supplémentaire est sa capacité à expliquer des prédictions basées sur des variables transformées (ex. composantes principales) en utilisant des caractéristiques originales, plus intuitives. Cette séparation entre modélisation et interprétation est particulièrement utile dans des contextes exigeant une forte lisibilité.

Limites

Malgré ses atouts, LIME présente plusieurs limites notables.

La définition du voisinage autour de l'instance à expliquer, surtout pour les données tabulaires, reste un défi. Aucune méthode standard n'existe et le choix des paramètres, comme la largeur du noyau, nécessite des tests itératifs et une vérification manuelle pour garantir la cohérence des explications.

En outre, l'échantillonnage de LIME, basé sur une loi gaussienne, ignore les corrélations entre variables. Cela peut générer des instances synthétiques irréalistes, altérant la qualité des interprétations.

La complexité du modèle explicatif (par exemple, la profondeur d'un arbre) doit être fixée a priori, ce qui implique un compromis entre clarté et fidélité à la prédiction originale.

LIME souffre aussi d'instabilité : deux instances proches ou des exécutions répétées peuvent produire des explications différentes, en raison de l'aléa dans l'échantillonnage. Ce manque de reproductibilité limite la confiance que l'on peut accorder aux résultats.

Enfin, les explications peuvent être manipulées. Un utilisateur mal intentionné pourrait ajuster les paramètres pour masquer certains biais du modèle, soulevant des problèmes éthiques et remettant en cause la fiabilité de la méthode.

2.2.3 SHAP (SHapley Additive exPlanations)

SHAP (SHapley Additive exPlanations) vise à expliquer localement les prédictions d'un modèle en attribuant à chaque caractéristique une contribution précise à la sortie du modèle pour une instance donnée. Cette méthode s'inspire des valeurs de Shapley issues de la théorie des jeux coopératifs, où chaque caractéristique joue le rôle d'un joueur participant à une coalition.

Dans ce cadre, chaque caractéristique contribue à la prédiction ou en est absente (remplacée par une valeur de référence). La prédiction finale est alors vue comme le résultat de toutes les combinaisons possibles de caractéristiques (coalitions).

Une innovation majeure de SHAP est la formulation de l'explication sous forme d'un modèle linéaire additif :

$$g(z') = \phi_0 + \sum_{i=1}^{M} \phi_i z'_i \tag{2.2}$$

où :
- $g(z')$ est le modèle explicatif,
- $z' \in \{0,1\}^M$ indique la présence (1) ou l'absence (0) des caractéristiques,
- ϕ_i représente la contribution (valeur de Shapley) de la caractéristique i,
- ϕ_0 est la prédiction moyenne (baseline).

Cette approche permet une interprétation unifiée applicable à différents types de données, notamment les images via des superpixels ou les textes via des mots-clés.

Propriétés Axiomatiques des Valeurs SHAP

Les valeurs SHAP héritent des propriétés fondamentales des valeurs de Shapley :

Précision locale (Local Accuracy)

La somme des contributions attribuées aux caractéristiques, ajoutée à la baseline, égale exactement la prédiction du modèle :

$$f(x) = \phi_0 + \sum_{i=1}^{M} \phi_i \tag{2.3}$$

Cela garantit que toute la "valeur" de la prédiction est répartie entre les caractéristiques.

Absence (Missingness)

Une caractéristique absente d'une coalition a une contribution nulle :

$$z'_i = 0 \Rightarrow \phi_i = 0 \tag{2.4}$$

Cette propriété assure la cohérence des explications lorsque certaines variables sont manquantes ou constantes.

Cohérence (Consistency)

Si une caractéristique apporte plus de contribution marginale dans un modèle f' qu'en f, sa valeur de Shapley ne doit pas diminuer :

$$\forall S \subseteq F \setminus \{i\}, \quad f_{S\cup\{i\}} - f_S \leq f'_{S\cup\{i\}} - f'_S \Rightarrow \phi_i^f \leq \phi_i^{f'} \tag{2.5}$$

Ainsi, l'importance relative d'une caractéristique reflète fidèlement son influence sur la prédiction.

Estimation des Valeurs SHAP

Le calcul exact des valeurs de Shapley est exponentiellement coûteux (2^M coalitions), rendant nécessaire l'utilisation de méthodes d'estimation efficaces.

KernelSHAP

KernelSHAP est une méthode agnostique au modèle proposée par Lundberg et Lee (2017). Elle suit les étapes suivantes :

- Génération aléatoire de vecteurs de coalition $z' \in \{0,1\}^M$,
- Reconstruction d'instances réelles $x_{z'}$ en remplaçant les caractéristiques absentes par des valeurs de fond,
- Évaluation du modèle sur ces instances : $f(x_{z'})$,
- Pondération des résultats selon un noyau spécifique :

$$\pi(z') = \frac{(M-1)}{\binom{M}{|z'|} \cdot |z'| \cdot (M - |z'|)} \tag{2.6}$$

- Régression pondérée pour estimer les ϕ_i.

Malgré sa flexibilité, KernelSHAP présente quelques limites :

- Hypothèse d'indépendance entre caractéristiques,
- Sensibilité aux interactions complexes,
- Coût computationnel élevé pour les données de grande dimension.

Des variantes conditionnelles ont été développées pour mieux capturer les dépendances structurelles des données, utiles notamment en médecine.

TreeSHAP

Spécialement conçu pour les modèles arborescents (arbres de décision, forêts aléatoires, boosting), TreeSHAP calcule les valeurs SHAP de manière exacte et rapide. Deux versions principales sont disponibles :

- *Interventional SHAP* : correspond aux valeurs de Shapley classiques,

- *Path-dependent SHAP* : intègre les chemins d'exécution de l'arbre pour estimer des contributions conditionnelles.

La complexité descend de $O(2^M)$ à $O(TLD^2)$, avec T le nombre d'arbres, L le nombre maximal de feuilles et D la profondeur maximale.

Méthode par permutations

La méthode par permutations constitue une alternative rapide et model-agnostic. Elle consiste à :

- Générer des permutations aléatoires des caractéristiques,
- Calculer les contributions marginales lors de leur ajout séquentiel,
- Moyenniser ces contributions pour obtenir les valeurs de Shapley.

Elle utilise souvent un échantillonnage antithétique (permutation directe et inverse) pour améliorer la stabilité :

$$\phi_i = \frac{1}{K} \sum_{k=1}^{K} \left(\Delta_i^{\pi_k} + \Delta_i^{\pi_k^{-1}} \right) \tag{2.7}$$

où π_k et π_k^{-1} sont une permutation et sa version inversée, et Δ_i la contribution marginale de la caractéristique i.

2.2.4 Méthodes Par Gradients

Les méthodes d'explicabilité fondées sur le gradient offrent une approche intuitive pour comprendre l'influence locale des variables d'entrée sur la sortie du modèle. Elles s'appuient sur le calcul des dérivées partielles de la fonction apprise par le réseau pour identifier les composantes de l'entrée qui contribuent significativement à une prédiction donnée.

L'analyse des gradients permet de sonder la structure locale de la fonction de décision et d'expliquer pourquoi un modèle a abouti à une certaine sortie pour une instance donnée. Ces approches sont particulièrement pertinentes dans des contextes médicaux, où il est crucial de comprendre les raisons d'une prédiction automatisée, que ce soit à partir d'images médicales (radiographies, IRM, images dermatologiques) ou de données tabulaires (dossiers cliniques, résultats biologiques, etc.).

Parmi les techniques les plus répandues, on distingue les *Saliency Maps* (Simonyan et al., 2013), les *Integrated Gradients* (Sundararajan et al., 2017), et *Grad-CAM* (Selvaraju et al., 2017).

Saliency Maps (Simonyan et al., 2013)

La méthode des cartes de saillance (*Saliency Maps*) repose sur une idée simple : identifier, pour une instance donnée, les entrées qui influencent le plus la sortie du modèle en observant la variation de la sortie par rapport à une variation infinitésimale de l'entrée. Mathématiquement, pour une entrée x et une fonction de sortie $f(x)$, on calcule le gradient :

$$S(x) = \left| \frac{\partial f(x)}{\partial x} \right|$$

Ce vecteur d'importance peut être visualisé sous forme d'image (dans le cas des données visuelles) ou interprété comme un vecteur d'attribution pour chaque variable (dans le cas des données tabulaires). Plus la dérivée est grande pour une composante, plus celle-ci est jugée influente.

Cependant, cette méthode présente plusieurs limites :

— elle est sensible au bruit et aux irrégularités locales de la fonction ;
— elle repose uniquement sur une information ponctuelle (gradient en un point), sans considérer le comportement global ;
— elle peut être instable dans des réseaux profonds non linéaires.

Malgré ces limites, sa simplicité et son faible coût computationnel en font une méthode de référence, notamment dans les applications médicales en phase exploratoire.[24]

Integrated Gradients (Sundararajan et al., 2017)

Les gradients intégrés constituent une amélioration formelle des cartes de saillance. L'objectif est de fournir une attribution fidèle et stable en intégrant les gradients sur un chemin reliant une entrée de référence x' à l'entrée réelle x :

$$IG_i(x) = (x_i - x'_i) \cdot \int_0^1 \frac{\partial f(x' + \alpha(x - x'))}{\partial x_i} \, d\alpha$$

Cette méthode respecte deux propriétés fondamentales :

— **Sensibilité** : si une variable modifie la sortie, elle doit recevoir une attribution non nulle ;
— **Complétude** : la somme des attributions doit correspondre à la différence de sortie entre l'entrée réelle et la référence.

Le choix de la référence est crucial :

— pour les images, on utilise souvent une image noire (tous les pixels à zéro) ;
— pour les données tabulaires, une moyenne ou un profil médical neutre est généralement sélectionné.

Les gradients intégrés sont particulièrement pertinents dans le domaine médical, car ils permettent une attribution plus robuste des variables cliniques ou des régions de l'image. Leur interprétabilité est améliorée par le fait qu'ils prennent en compte l'ensemble du chemin entre la base et l'entrée analysée.[28]

Grad-CAM (Selvaraju et al., 2017)

La méthode *Grad-CAM* est spécifiquement conçue pour les réseaux de neurones convolutifs utilisés en vision par ordinateur. Elle permet d'identifier les régions spatiales d'une image qui ont contribué à une prédiction donnée, en exploitant les cartes d'activation d'une couche convolutionnelle intermédiaire.

Le mécanisme repose sur l'utilisation des gradients pour calculer un poids d'importance α_k pour chaque carte d'activation A^k :

$$\alpha_k = \frac{1}{Z} \sum_{i,j} \frac{\partial f(x)}{\partial A^k_{i,j}}$$

Puis la carte Grad-CAM est obtenue par une combinaison linéaire pondérée :

$$L_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k A^k \right)$$

L'application de la fonction ReLU permet de ne conserver que les contributions positives à la classe cible. Le résultat est une carte thermique que l'on peut superposer à l'image d'origine pour localiser visuellement les zones influentes.

Cette méthode est largement utilisée dans le domaine médical pour interpréter des modèles CNN appliqués à la radiologie, la dermatologie, ou l'imagerie histopathologique. Elle permet aux médecins de vérifier si l'attention du modèle est focalisée sur des régions cliniquement pertinentes.[23]

Analyse comparative

- Les **Saliency Maps** sont rapides et simples, mais sensibles au bruit et peu robustes.
- Les **Integrated Gradients** apportent une meilleure stabilité mathématique, au prix d'un coût de calcul plus élevé.
- **Grad-CAM** offre une interprétation spatiale efficace, particulièrement adaptée aux données visuelles médicales.

Ces approches ne s'excluent pas mutuellement ; au contraire, leur combinaison peut fournir une vision plus complète de la prise de décision d'un modèle, surtout dans des contextes sensibles comme la santé, où les enjeux d'interprétabilité sont cruciaux pour la validation clinique, l'identification des biais et l'acceptabilité par les praticiens.

2.3 Méthodes d'explicabilité globales

Les méthodes d'explicabilité globale cherchent à décrire le comportement général d'un modèle en analysant l'influence des variables sur l'ensemble des prédictions. Contrairement aux approches locales, elles offrent une vision d'ensemble, utile pour mieux comprendre les tendances apprises par le modèle.

Dans cette section, nous présentons quatre méthodes représentatives : l'importance des variables (PFI), les graphiques de dépendance partielle (PDP), les effets accumulés localement (ALE) et les modèles substituts globaux. Chacune d'elles apporte un éclairage complémentaire sur la manière dont le modèle exploite les caractéristiques d'entrée.

2.3.1 Importance des caractéristiques par permutation (Permutation Feature Importance – PFI)

L'importance des caractéristiques par permutation est une méthode intuitive et largement utilisée pour quantifier la contribution de chaque variable d'entrée à la performance d'un modèle d'apprentissage automatique. Elle repose sur une idée simple : si la permutation aléatoire des valeurs d'une variable entraîne une dégradation significative de la performance prédictive, alors cette variable est considérée comme importante pour le modèle.[17]

Principe général

L'importance d'une caractéristique est évaluée en mesurant l'augmentation de l'erreur du modèle lorsque les valeurs de cette caractéristique sont aléatoirement réarrangées dans le jeu de données. Cette permutation rompt la relation entre la variable et la cible, ce qui permet d'estimer dans quelle mesure le modèle dépend de cette information pour

prédire correctement. Si l'erreur augmente fortement, la caractéristique est jugée importante ; si l'erreur reste stable, cela signifie que le modèle ne l'exploitait pas de manière significative.[17]

Origine théorique

La méthode PFI a été introduite par Leo Breiman (2001) dans le contexte des forêts aléatoires. Plus récemment, Fisher, Rudin et Dominici (2019) ont proposé une généralisation indépendante du modèle, applicable à tout algorithme d'apprentissage. Leur approche a permis de clarifier la distinction entre importance marginale et importance conditionnelle des caractéristiques, et de mieux formaliser la notion d'utilisation effective d'une variable par un modèle prédictif.

Algorithme

L'algorithme de base pour estimer la PFI est le suivant :

1. Estimer la performance de référence du modèle sur un jeu de test (par exemple, erreur quadratique moyenne, erreur absolue moyenne, ou perte logarithmique).
2. Pour chaque variable X_j :
   — Permuter aléatoirement les valeurs de X_j dans le jeu de test, tout en laissant les autres variables inchangées.
   — Réévaluer la performance du modèle sur ce jeu de données modifié.
   — Calculer l'importance de X_j comme la différence (ou le ratio) entre la performance modifiée et la performance de référence.

Ce processus peut être répété plusieurs fois afin de réduire la variance de l'estimation. Une moyenne des valeurs obtenues est ensuite calculée, et les résultats peuvent être représentés graphiquement avec des intervalles de confiance.

Utilisation avec des métriques inversées

Lorsque l'on travaille avec des métriques où des valeurs plus élevées sont préférables (comme la précision ou l'AUC), il convient d'inverser les signes dans le calcul des différences ou des ratios de performance pour maintenir la cohérence de l'interprétation.

Importance conditionnelle vs importance marginale

L'un des principaux défis de la méthode PFI réside dans la présence de corrélations entre les caractéristiques. Lorsque les variables sont interdépendantes, la permutation aléatoire peut générer des combinaisons irréalistes ou non observées dans les données réelles (ex. : un patient avec un IMC de 40 mais une tension artérielle très basse). Ce problème affecte l'interprétation des résultats, car l'augmentation de l'erreur ne reflète plus uniquement la perte d'information de la variable permutée, mais aussi la rupture de relations naturelles entre variables.

Pour atténuer cet effet, on peut recourir à une estimation conditionnelle de l'importance, qui repose sur un échantillonnage à partir de la distribution conditionnelle $P(X_j \mid X_{-j})$, plutôt que marginale. Plusieurs approches sont possibles :

— Segmenter les données en sous-groupes homogènes (par ex. selon des variables corrélées), et estimer l'importance de manière locale (Molnar et al., 2023).

— Utiliser des méthodes d'imputation conditionnelle (Fisher et al., 2019) ou des générateurs de données réalistes (Watson & Wright, 2021).
— Pour les forêts aléatoires, employer des variantes spécifiques intégrant des permutations conditionnelles (Debeer & Strobl, 2020).

Exemple appliqué aux données médicales

Prenons l'exemple d'un modèle de classification entraîné à prédire la présence de diabète à partir de données cliniques issues du Pima Indian Diabetes Dataset. Supposons que le modèle utilise des variables comme la glycémie, le nombre de grossesses et l'IMC. En permutant uniquement la variable « glycémie », on observe une nette augmentation de l'erreur prédictive. Cela signifie que le modèle dépend fortement de cette variable pour effectuer une prédiction fiable.

Inversement, si la permutation du nombre de grossesses n'entraîne aucun changement significatif, cela suggère que cette variable n'a que peu d'impact sur la décision finale du modèle, dans le contexte de ces données.

Dans un autre exemple visuel, imaginons un modèle de CNN entraîné sur des images de lésions cutanées (comme le dataset DermaMNIST). En perturbant les cartes de caractéristiques associées à certaines zones de l'image (par exemple la texture ou la bordure), et en observant la dégradation de la précision, on peut estimer l'importance de ces éléments visuels dans la prédiction.

Avantages de la PFI

— Interprétation simple : La PFI mesure directement la contribution d'une variable à la performance du modèle.
— Méthode globale : Elle fournit une vue d'ensemble du rôle des caractéristiques, indépendamment de la structure interne du modèle.
— Compatibilité universelle : Applicable à tout modèle de type boîte noire, sans nécessiter de modification de l'algorithme.
— Prise en compte des interactions : En permutant une variable, on détruit aussi ses interactions avec d'autres variables, ce qui permet de mesurer leur effet combiné.
— Pas de réentraînement nécessaire : La méthode est rapide car elle ne nécessite pas d'entraîner plusieurs versions du modèle.

Limites de la PFI

— Biais dû aux corrélations : Lorsque des variables sont fortement corrélées, la permutation peut produire des instances incohérentes, biaisant ainsi la mesure d'importance.
— Perte de granularité : La PFI ne précise pas comment une variable influence la prédiction, ni dans quel sens (positif ou négatif).
— Sensibilité au surapprentissage : Si le modèle est surajusté, il peut attribuer une importance artificiellement élevée à des variables non pertinentes.
— Dépendance à la métrique d'erreur : La PFI mesure uniquement l'impact sur la performance, et non la sensibilité de la prédiction à la variation des entrées.
— Variance élevée : Les résultats peuvent varier d'une permutation à l'autre, d'où la nécessité de moyennes et d'intervalles de confiance.

— Interprétation parfois difficile : En présence de fortes interactions, la somme des importances ne correspond pas à la perte totale d'information, car les effets d'interaction sont répartis sur plusieurs variables.

Bonnes pratiques

— Toujours évaluer la PFI sur un jeu de données indépendant (validation ou test), afin d'éviter les biais induits par un surajustement éventuel sur les données d'entraînement.
— Compléter l'analyse avec d'autres méthodes, telles que les valeurs SHAP (pour une interprétation locale) ou les PDP (pour visualiser les effets marginaux).
— En présence de corrélations fortes, privilégier les estimations conditionnelles ou les regroupements en sous-populations homogènes.

2.3.2 Partial Dependence Plots(PDP)

Dans le contexte de l'explicabilité des modèles de type « boîte noire » tels que les *réseaux de neurones*, les *Partial Dependence Plots* (PDP) constituent un outil d'analyse globale permettant d'explorer la manière dont une ou plusieurs variables d'entrée influencent la prédiction moyenne du modèle. L'intérêt de cette approche réside dans sa capacité à restituer, de manière intuitive, **l'effet marginal d'une caractéristique d'entrée** sur la sortie du réseau, tout en **moyennant sur les autres variables**.

L'idée sous-jacente consiste à simuler, pour une valeur donnée d'une variable cible, la prédiction du modèle en remplaçant cette valeur dans toutes les observations du jeu de données, puis à en faire la moyenne. Le résultat est un graphique représentant la relation entre cette variable et la prédiction du modèle, permettant d'évaluer si l'impact est croissant, décroissant, linéaire ou non linéaire.[17]

Formulation mathématique

Soit $f(\mathbf{x})$ un réseau de neurones entraîné, et $\mathbf{x} = (x_1, x_2, ..., x_p)$ un vecteur d'entrée. Le graphique de dépendance partielle pour une variable x_j est défini comme :

$$PD(x_j) = \mathbb{E}_{\mathbf{x}_{-j}}[f(x_j, \mathbf{x}_{-j})]$$

où $\mathbf{x}_{-j}$ représente toutes les autres variables que x_j. En pratique, cette espérance est approximée par une moyenne empirique sur un jeu de données contenant n observations :

$$\hat{PD}(x_j) = \frac{1}{n} \sum_{i=1}^{n} f(x_j, \mathbf{x}_{-j}^{(i)})$$

Application aux données médicales non visuelles

Dans le cas de *données tabulaires* issues de dossiers médicaux (comme les jeux de données `Breast Cancer Wisconsin` ou `Pima Indians Diabetes`), les PDP permettent de visualiser l'effet moyen de variables cliniques telles que le *rayon moyen d'une cellule*, le *niveau de glucose sanguin* ou encore l'*âge du patient* sur la prédiction du modèle neuronal.

Par exemple, un PDP construit pour la variable `radius_mean` dans un réseau de neurones multicouche (*MLP*) destiné à prédire la malignité d'une tumeur peut révéler que des valeurs croissantes de cette variable sont associées à une probabilité plus élevée

de cancer malin. Ce type de représentation facilite l'interprétation globale du modèle en mettant en lumière les tendances générales apprises, ce qui est essentiel dans des contextes cliniques où la transparence des décisions est primordiale.

Application aux données médicales visuelles

Dans le cadre de *l'imagerie médicale*, les PDP peuvent être appliqués indirectement à des caractéristiques apprises par un *réseau de neurones convolutif* (CNN). Par exemple, en extrayant les activations d'une couche intermédiaire (c'est-à-dire des caractéristiques latentes), il est possible d'étudier l'influence de ces attributs visuels (comme la texture, l'intensité ou la régularité) sur la prédiction finale du modèle.

Sur un jeu de données comme `DermaMNIST`, un PDP appliqué à une *caractéristique latente* correspondant à la rugosité d'une lésion cutanée pourrait montrer que, plus cette caractéristique est marquée, plus la probabilité de prédiction d'un mélanome est élevée. Cela permet aux cliniciens de mieux comprendre quels éléments visuels influencent le diagnostic automatisé, renforçant ainsi la confiance dans les décisions du modèle.

Intérêt pour l'explicabilité des réseaux de neurones

Les PDP jouent un rôle important dans l'explicabilité globale des réseaux de neurones en fournissant une vue d'ensemble du comportement du modèle vis-à-vis de chaque variable. Ils permettent notamment de :

- détecter des **relations non linéaires** entre les variables et la sortie du modèle ;
- vérifier la **cohérence clinique** des comportements appris ;
- servir de support visuel dans le cadre d'une **collaboration interdisciplinaire** (par exemple avec des radiologues ou médecins spécialistes).

Limites

Malgré leur simplicité d'interprétation, les PDP présentent plusieurs limites, particulièrement critiques dans le domaine médical :

- **Supposition d'indépendance** : le PDP suppose que la variable analysée est indépendante des autres, ce qui est rarement le cas dans les données médicales. Cela peut engendrer des combinaisons irréalistes et biaiser les résultats.
- **Effets locaux masqués** : comme le PDP fournit une moyenne globale, il peut occulter des comportements spécifiques à certains sous-groupes de patients (par exemple, différences selon le sexe, l'âge ou l'origine ethnique).
- **Représentations abstraites** : dans les réseaux profonds, les variables intermédiaires sont souvent abstraites, ce qui peut rendre leur interprétation difficile sans expertise complémentaire.

Compléments et alternatives Pour surmonter ces limites, d'autres méthodes plus robustes peuvent être utilisées en complément, telles que :

- les **graphes ICE** (*Individual Conditional Expectation*), qui révèlent les effets à l'échelle individuelle ;
- les **Accumulated Local Effects (ALE)**, qui tiennent compte de la distribution conditionnelle des données ;
- ou encore les **méthodes basées sur les gradients** (par exemple, *Integrated Gradients*, *Grad-CAM*) pour une analyse plus fine des réseaux convolutifs.

2.3.3 Accumulated Locale Effects (ALE)

L'idée principale d'ALE est d'étudier l'impact d'une caractéristique (par exemple, l'âge d'un patient ou une valeur biologique) sur la prédiction du modèle (comme le risque de développer une maladie), tout en prenant en compte les autres variables présentes.

Plutôt que de modifier une seule variable en ignorant les autres (ce qui pourrait donner des résultats peu réalistes), ALE examine de petites variations locales de cette variable tout en gardant les autres constantes pour chaque individu. En accumulant ces petits effets locaux, on obtient une courbe globale qui montre comment la variable influence la prédiction en moyenne.

Fonctionnement simplifié

1. **Découpage par intervalles** : La variable d'intérêt est divisée en plusieurs segments (par exemple, tranches d'âge). 2. **Calcul des différences** : Pour chaque segment, on observe comment la prédiction change légèrement quand on fait varier la variable d'intérêt. 3. **Somme cumulative** : Ces variations sont additionnées progressivement pour obtenir l'effet global de la variable sur la sortie du modèle.

Le résultat est une courbe qui montre clairement si une augmentation de la variable tend à augmenter ou diminuer la prédiction du modèle.

Pourquoi utiliser ALE ?

ALE présente plusieurs avantages :

- **Prend en compte les corrélations** : contrairement à certaines méthodes comme les PDP, ALE ne suppose pas que les variables sont indépendantes, ce qui évite des interprétations biaisées.
- **Adapté aux modèles complexes** : il fonctionne bien avec les réseaux de neurones profonds et autres modèles opaques.
- **Interprétation intuitive** : la courbe ALE est facile à lire : au-dessus de zéro, la variable favorise la prédiction ; en dessous, elle la limite.

Limites

- **Effets locaux** : L'interprétation reste locale à chaque intervalle, donc la courbe ne donne pas toujours une vue complète du comportement global du modèle.
- **Complexité croissante avec les interactions** : analyser l'effet combiné de deux variables devient plus difficile à interpréter.
- **Dépendance à la structure des données** : lorsque les variables sont très corrélées, il peut être compliqué de distinguer leur effet individuel.

Exemples d'applications médicales

- **Imagerie médicale** : Sur un réseau neuronal analysant des radiographies, ALE peut montrer quelles zones ouquelles intensités influencent le diagnostic, par exemple la détection d'une tumeur.
- **Données cliniques** : Avec des données de patients (âge, poids, antécédents...), ALE aide à comprendre quels facteurs jouent le plus sur la prédiction d'un risque cardiaque ou d'une complication post-opératoire.

En résumé, ALE est un outil puissant pour explorer finement l'impact des variables dans des modèles complexes, tout en tenant compte des réalités des données, notamment en médecine où les corrélations sont fréquentes.

2.3.4 Modèles substituts globaux (Global Surrogate Models)

Les *modèles substituts globaux* constituent une approche d'explicabilité qui consiste à approximer un modèle complexe, souvent considéré comme une boîte noire (tel qu'un réseau de neurones profond), à l'aide d'un modèle interprétable. L'objectif est de capturer le comportement global du modèle initial de manière intelligible, sans nécessairement altérer ses performances prédictives.[17]

Principe général

Un modèle substitut (ou *surrogate model*) est un modèle simple g, tel qu'un arbre de décision, une régression linéaire ou un modèle additif, entraîné à imiter les prédictions d'un modèle complexe f. Ce substitut est conçu pour être globalement interprétable, c'est-à-dire fournir une compréhension d'ensemble du comportement du modèle f sur tout l'espace des données d'entrée $\mathcal{X}$.

Formellement, on considère un jeu de données $\mathcal{D} = \{x^{(i)}\}_{i=1}^{n}$, où chaque $x^{(i)}$ est une instance d'entrée, et $f(x^{(i)})$ la prédiction associée du modèle complexe. Le modèle substitut g est alors appris en minimisant une fonction de perte $\mathcal{L}$ entre les prédictions de g et celles de f :

$$g = \arg\min_{g \in \mathcal{G}} \sum_{i=1}^{n} \mathcal{L}(f(x^{(i)}), g(x^{(i)}))$$

où $\mathcal{G}$ représente la famille des modèles interprétables (par exemple, les arbres de décision de faible profondeur).

Objectif d'interprétabilité

Contrairement à une approche traditionnelle où le modèle interprétable est directement entraîné sur les étiquettes réelles y, ici le modèle g apprend à prédire les sorties du modèle f. L'intérêt est double :

- obtenir une vue simplifiée du modèle f,
- tout en conservant une fidélité suffisante aux prédictions de f, ce qui permet une interprétation approximative mais pertinente.

Évaluation de la fidélité

La qualité de l'approximation est mesurée par la *fidelity* (fidélité), qui quantifie dans quelle mesure le modèle substitut g reproduit fidèlement les prédictions du modèle complexe f. On peut l'évaluer sur un jeu de test $\mathcal{D}_{\text{test}}$ comme suit :

$$\text{Fidelity}(g, f) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{x \in \mathcal{D}_{\text{test}}} \mathbb{1}\{g(x) = f(x)\}$$

ou, dans le cas de régression :

$$\text{Fidelity}(g, f) = \frac{1}{|\mathcal{D}_{\text{test}}|} \sum_{x \in \mathcal{D}_{\text{test}}} (g(x) - f(x))^2$$

Une faible fidélité indique que le modèle substitut ne reflète pas bien le comportement de f, tandis qu'une fidélité élevée suggère que g constitue une approximation globale acceptable.

Application à l'explicabilité des réseaux de neurones

Dans le contexte des réseaux de neurones profonds, souvent utilisés pour des tâches complexes comme la classification d'images médicales ou la prédiction de diagnostics, les modèles substituts permettent d'en résumer les comportements sans devoir inspecter directement leurs paramètres internes (poids, activations, etc.).

Par exemple, un réseau convolutif f entraîné sur le dataset DermaMNIST peut être approché par un arbre de décision g qui capture les principales règles de décision apprises par le réseau (ex. : « Si la couleur dominante est sombre et les bords sont irréguliers, alors la lésion est classée comme maligne »).

De même, dans un dataset non visuel comme le Breast Cancer Wisconsin dataset, un modèle substitut peut mettre en évidence que le réseau de neurones base majoritairement ses décisions sur des variables comme la concavité des contours ou la texture moyenne.

Avantages des modèles substituts

— **Interprétation globale** : Fournit une vue d'ensemble compréhensible du comportement du modèle complexe sur l'ensemble des données.
— **Flexibilité** : Peut être utilisé avec n'importe quel modèle prédictif, quelle que soit sa complexité ou son opacité.
— **Indépendance du modèle original** : Le modèle substitut n'a pas besoin d'accéder aux gradients ou à la structure interne du modèle initial.

Limites

— **Approximation partielle** : Le modèle substitut peut omettre des subtilités importantes si la complexité du modèle original est trop élevée.
— **Risque de mauvaise interprétation** : Une faible fidélité peut induire en erreur sur le comportement réel du modèle f.
— **Choix du modèle substitut** : Il n'existe pas de méthode universelle pour choisir le bon type de modèle substitut ; un compromis entre interprétabilité et fidélité est souvent nécessaire.

Bonnes pratiques

— Toujours mesurer la fidélité pour s'assurer que l'explication est représentative.
— Tester différents modèles substituts pour évaluer celui qui capture le mieux les comportements clés du modèle cible.
— Compléter l'approche globale avec des méthodes locales (LIME, SHAP) pour analyser des prédictions spécifiques.

2.4 Méthodes Avancées

Certaines approches d'explicabilité s'appuient sur des fondements mathématiques plus poussés afin d'analyser en profondeur le comportement interne des modèles. Bien qu'elles soient moins courantes en pratique, elles offrent un cadre théorique rigoureux pour l'interprétation.

Dans cette section, nous présentons la méthode de décomposition de Taylor, qui permet d'attribuer de manière analytique les contributions des variables d'entrée à la prédiction d'un réseau de neurones.

2.4.1 Deep Taylor Decomposition

La méthode *Deep Taylor Decomposition* (DTD) a été conçue pour expliquer les décisions des réseaux de neurones, en particulier ceux utilisant des fonctions d'activation de type ReLU. Elle repose sur une décomposition de Taylor appliquée de manière récursive à travers les couches du réseau, afin de redistribuer la prédiction finale vers les neurones internes, puis jusqu'aux entrées, sous forme de *scores de pertinence* (*relevance scores*).[25]

Objectif de la méthode

DTD cherche à répondre à la question : *quelles caractéristiques d'entrée ont le plus contribué à une prédiction donnée du modèle ?* Cette approche est particulièrement utile dans le domaine médical, par exemple pour :

- Visualiser les zones clés d'une image IRM ou cutanée à l'origine de la prédiction ;
- Identifier les variables cliniques déterminantes (comme la glycémie ou l'IMC) dans un modèle prédictif.

Fondements mathématiques

Considérons un réseau de neurones f composé de n couches avec activations ReLU :

$$f = f_n \circ f_{n-1} \circ \cdots \circ f_1$$

où chaque couche f_l applique une opération linéaire suivie d'une activation ReLU :

$$f_l(a_l) = \max(0, W_l a_l)$$

L'idée est de propager la prédiction finale $f(x)$, notée R, en sens inverse à travers le réseau. On applique un développement de Taylor du premier ordre autour d'un *point de référence* (ou *root point*) $\tilde{a}_l$:

$$R^l(a_l) \approx \nabla_{a_l} R^{l+1}(f_l(\tilde{a}_l)) \cdot (a_l - \tilde{a}_l)$$

Ce développement permet de redistribuer R sous forme de scores de pertinence vers les activations précédentes. Le processus est répété jusqu'à la couche d'entrée, produisant une attribution de pertinence pour chaque caractéristique (pixel, variable clinique...).

Rôle du point de référence (*root point*)

Le choix du point $\tilde{a}_l$ est crucial. Il existe plusieurs options :

- Un point constant (ex. zéro), auquel cas la méthode devient équivalente à *gradient* $\times$ *input*.
- Un point dépendant dynamiquement de l'entrée x, mais ce choix peut rendre la méthode non falsifiable, c'est-à-dire que toute explication peut être justifiée a posteriori.

L'étude rigoureuse de Sixt et Landgraf (2022) montre que, dans la pratique, les root points choisis se situent souvent en dehors de la région linéaire locale des ReLU, ce qui viole les hypothèses du développement de Taylor. Par conséquent, les scores de pertinence obtenus peuvent ne pas être mathématiquement valides.

Expériences empiriques

Les auteurs ont testé la DTD sur divers réseaux ReLU et ont constaté :

- Une équivalence fréquente avec la méthode *gradient* $\times$ *input*.
- Des violations des conditions de validité mathématique du développement de Taylor.
- Des cartes de pertinence potentiellement trompeuses.

Applications aux données médicales

Images médicales (données visuelles) : Sur des données comme DermaMNIST ou des mammographies, la DTD attribue des scores de pertinence aux pixels. Cependant, si le root point est mal choisi, la carte produite ne reflète pas nécessairement les régions médicalement pertinentes.

Données cliniques tabulaires : Sur des jeux comme *Pima Indian Diabetes* ou *Breast Cancer Wisconsin*, la méthode permet de quantifier la contribution des variables cliniques. Là encore, les violations des hypothèses peuvent biaiser l'interprétation.

Avantages

- Permet une attribution hiérarchique et continue des scores de pertinence.
- Ne perturbe pas les entrées (contrairement à LIME ou SHAP).
- Compatible avec la structure interne du réseau (via rétropropagation).

Limites (selon Sixt & Landgraf, 2022)

- Forte dépendance au choix du root point.
- Équivalence fréquente à *gradient* $\times$ *input*, sans réelle valeur ajoutée.
- Absence de garantie mathématique lorsque les hypothèses sont violées.
- Possibilité d'explications incohérentes ou biologiquement non pertinentes.

2.5 Comparaison des méthodes d'explicabilité

Au terme de cette comparaison, il apparaît que chaque méthode d'explicabilité apporte une forme d'éclairage complémentaire sur le comportement du modèle. Les approches

locales, comme LIME, SHAP ou les méthodes par gradient (Saliency Maps, Grad-CAM, Integrated Gradients), permettent de comprendre les décisions prises pour une instance donnée, en identifiant les variables les plus influentes. Elles sont particulièrement utiles pour détecter des cas d'incohérence ou mieux interpréter des résultats inattendus. De leur côté, les méthodes globales comme les PDP, ALE, l'importance des variables ou les substituts globaux offrent une vue d'ensemble du modèle et facilitent l'analyse de son comportement moyen sur l'ensemble des données. Enfin, la méthode de Taylor, bien que plus théorique, illustre comment une approche mathématique locale peut aussi conduire à des interprétations précises. Ainsi, combiner plusieurs méthodes permet d'obtenir une compréhension plus complète et plus fiable du modèle, en croisant les points de vue locaux et globaux.

Méthode	**Portée**	**Principe**	**Avantages**	**Limites**
LIME	Locale	Ajuste un modèle simple autour d'une prédiction cible	Intuitif, compatible avec différents types de données	Sensible aux perturbations, dépend du voisinage
SHAP	Locale	Décompose la prédiction selon les valeurs de Shapley	Solide théoriquement, attribution équitable	Coût computationnel élevé
Saliency Map	Locale	Visualise les gradients par rapport à l'entrée (norme absolue)	Rapide, facile à implémenter	Résultats parfois bruités ou peu interprétables
Grad-CAM	Locale	Utilise les gradients des couches convolutionnelles pour localiser les régions importantes	Visualisation intuitive sur images	Limité aux CNN
Integrated Gradients	Locale	Moyenne des gradients entre une baseline et l'entrée réelle	Attribution stable, atténue les discontinuités des gradients	Sensible au choix de la baseline
Décomposition de Taylor	Locale	Approximation locale via un développement de Taylor d'ordre 1	Mathématiquement rigoureuse	Peu utilisée, nécessite la différentiabilité
Importance des variables	Globale	Perturbe chaque variable pour observer l'impact sur la performance	Simple, compatible avec tout modèle	Moins fiable si variables corrélées
PDP	Globale	Moyenne des prédictions en faisant varier une variable	Intuitif, visualise l'effet moyen	Suppose indépendance des variables
ALE	Globale	Moyenne des effets locaux dans des intervalles conditionnels	Corrige PDP pour variables corrélées	Moins intuitive, dépend du découpage
Substitut global	Globale	Modèle simple entraîné sur les prédictions du modèle complexe	Vue synthétique globale	Moins fidèle, simplification excessive

Table 2.5 – Comparaison des méthodes d'explicabilité utilisées

Synthèse du chapitre

Ce chapitre a présenté différentes méthodes d'explicabilité des réseaux de neurones, couvrant les approches locales (LIME, SHAP, Grad-CAM, Saliency Maps, Integrated Gradients), globales (PDP, ALE, modèles substituts) ainsi qu'une méthode avancée basée sur la décomposition de Taylor. Ces techniques permettent d'éclairer le rôle des variables ou des zones d'entrée dans les prédictions des modèles.

Pour le projet, les méthodes LIME et SHAP ont été retenues pour les explications locales sur données tabulaires, en raison de leur capacité à fournir des résultats compréhensibles et pertinents. Pour l'analyse d'images médicales, Grad-CAM a été privilégiée car elle offre une visualisation claire des régions influentes détectées par les réseaux convolutionnels. La méthode de Taylor, plus théorique, apporte une compréhension mathématique utile des contributions des variables.

Ces choix sont essentiels dans le domaine médical où la transparence des modèles est primordiale pour assurer la confiance des professionnels de santé. Une explicabilité adaptée facilite l'acceptation des résultats et contribue à leur intégration en pratique clinique. Ce cadre théorique ouvre la voie à l'implémentation et à l'évaluation concrète des méthodes dans le chapitre suivant

Chapitre 3

Etude des cas et application des méthodes d'explicabilité

3.1 Introduction

Ce chapitre a pour objectif principal de mettre en pratique les différentes méthodes d'explicabilité étudiées précédemment, en les appliquant à des réseaux de neurones entraînés sur des données médicales. L'enjeu ici n'est pas seulement d'obtenir de bonnes performances de classification, mais surtout de comprendre *comment* les modèles prennent leurs décisions. L'interprétabilité devient alors un outil essentiel pour analyser, valider ou questionner les prédictions, en particulier dans des domaines sensibles comme la santé.

Afin de rendre cet objectif accessible et clair, le choix des données utilisées s'est orienté vers des **jeux de données simples, bien structurés et faciles à manipuler**. Pour la partie visuelle, deux sous-ensembles issus de la base *MedMNIST* ont été sélectionnés : **DermaMNIST**, qui contient des images dermatologiques associées à sept classes de pathologies, et **BreastMNIST**, qui vise la détection binaire du cancer du sein à partir d'images médicales. Ces jeux de données ont l'avantage d'être accessibles, standardisés, et adaptés à des expérimentations pédagogiques tout en restant ancrés dans des problématiques médicales réelles.

Concernant les données **non visuelles**, trois jeux bien connus ont été retenus : **Breast Cancer Wisconsin**, **Pima Indians Diabetes** et **Heart Disease (Cleveland)**. Ces bases de données, largement utilisées dans la littérature, permettent de travailler avec des variables cliniques classiques et sont particulièrement adaptées à l'application de modèles tabulaires simples comme les réseaux de neurones multicouches (MLP). Elles offrent un terrain propice pour tester des méthodes d'explicabilité locale et globale tout en conservant une bonne lisibilité des résultats.

Les approches mises en œuvre dans ce chapitre reposent donc sur deux types de modèles : des **réseaux convolutifs (CNN)** pour le traitement des images, et des **réseaux de neurones entièrement connectés (MLP)** pour les données tabulaires. L'explicabilité des modèles est étudiée à l'aide de plusieurs techniques complémentaires :

- **Saliency Map**, **Integrated Gradients** et **Grad-CAM** pour la visualisation des zones d'attention dans les images,
- **LIME**, **SHAP** et **Deep Taylor** pour l'analyse des contributions des variables dans les prédictions tabulaires,
- ainsi que la **Permutation Feature Importance**, utile pour l'évaluation globale de l'impact des variables d'entrée.

Le fil conducteur de ce chapitre est donc résolument pédagogique : plutôt que de viser des modèles très complexes ou ultra-performants, l'accent est mis sur la compréhension des résultats, la simplicité des architectures et la clarté des interprétations. L'objectif est de montrer concrètement comment les outils d'explicabilité permettent d'ouvrir la "boîte noire" des réseaux de neurones, et ce, dans des contextes proches d'applications médicales réelles.

3.2 Etude sur les données médicales visuelles

Dans le cadre de cette étude, une attention particulière est portée aux données médicales visuelles, qui jouent un rôle clé dans de nombreux diagnostics cliniques. Issues de techniques telles que la mammographie ou la dermoscopie, ces images permettent de mettre en évidence des anomalies souvent invisibles à l'œil nu. Leur richesse informationnelle en fait des supports de choix pour l'application de méthodes d'apprentissage profond, notamment les réseaux de neurones convolutifs.

Cette section présente les spécificités de ces données, les étapes de prétraitement nécessaires à leur exploitation, ainsi que les modèles utilisés pour leur analyse. L'objectif est de montrer comment ces images peuvent être transformées en vecteurs d'information exploitables par des modèles intelligents, tout en tenant compte des particularités médicales et techniques qu'elles impliquent.

3.2.1 Présentation des jeux de données utilisés

DermaMNIST

Le jeu de données *DermaMNIST* fait partie de la collection **MedMNIST v2**, une suite de jeux de données médicaux standardisés conçus pour l'expérimentation en apprentissage automatique. Il est dédié à la **classification d'images dermatologiques** et comprend un total de **10 015 images** en couleur (*RGB*), toutes redimensionnées à une résolution de **28×28 pixels**.

Ces images proviennent de la base de données **HAM10000 (Human Against Machine with 10000 training images)**, largement utilisée en dermatologie pour l'analyse automatique des lésions cutanées. Elles ont été annotées par des experts cliniciens et réparties en **sept classes distinctes** :

- Kératoses actiniques et carcinomes intraépithéliaux / maladie de Bowen
- Carcinomes basocellulaires
- Lésions bénignes de type kératosique
- Dermatofibromes
- Naevi mélanocytaires
- Mélanomes
- Lésions vasculaires

Les données sont organisées selon trois sous-ensembles : *entraînement*, *validation* et *test*, avec une répartition équilibrée dans la mesure du possible. Ce dataset permet d'évaluer la capacité des modèles à différencier des lésions visuellement proches, dans un contexte médical critique.[29]

BreastMNIST

Le jeu de données *BreastMNIST*, également issu de la collection **MedMNIST v2**, concerne la **classification binaire des masses mammaires** à partir d'images échographiques. Il comprend un total de **780 images** en niveaux de gris (*1 canal*), de taille **28×28 pixels**, représentant des coupes échographiques du sein.

Les images sont dérivées de l'étude *"A Breast Ultrasound Dataset for Systematic Classification"* (Al-Dhabyani et al., 2020), qui proposait initialement une base de données échographiques annotées. Pour BreastMNIST, les images ont été regroupées en deux classes :

— **Classe 0** : masse bénigne
— **Classe 1** : masse maligne

Les étiquettes sont basées sur des diagnostics confirmés, incluant dans certains cas des résultats histopathologiques. Ce format réduit (28×28) permet une expérimentation rapide tout en conservant les informations discriminantes nécessaires à la classification.

Comme pour DermaMNIST, les données sont divisées en trois sous-ensembles : *train*, *validation* et *test*, assurant une structure cohérente pour l'évaluation des modèles.[1]

3.2.2 Prétraitement et exploration des données

Les jeux de données visuelles `DermaMNIST` et `BreastMNIST` ont été prétraités selon une procédure standardisée afin de les rendre exploitables par les modèles de réseaux de neurones convolutifs.

Étapes de prétraitement communes

Pour chaque dataset, les étapes suivantes ont été appliquées :

— **Téléchargement automatisé** à l'aide de l'API `medmnist`, pour les trois ensembles : entraînement, validation et test.
— **Conversion au format `ImageFolder`** : les images initialement encodées dans des fichiers `.npz` ont été extraites, regroupées par classe dans une arborescence de dossiers, puis enregistrées au format `.png`.
— **Transformation des images** en tenseurs normalisés via `ToTensor()`, assurant leur compatibilité avec `PyTorch`.

Cette procédure garantit une uniformisation du format d'entrée, tout en préservant les caractéristiques visuelles essentielles à la tâche de classification.

Visualisation des données

Des visualisations ont été réalisées afin d'inspecter la qualité des images et la représentation visuelle des différentes classes pour chaque dataset.

Pour le dataset `BreastMNIST`, qui se concentre sur la classification entre images avec ou sans masse tumorale, deux sous-ensembles visuels distincts ont été illustrés.

Cette visualisation met en évidence le contraste entre les deux classes, souvent ténu, et souligne les défis liés à la détection automatique de masses mammaires à partir d'images médicales

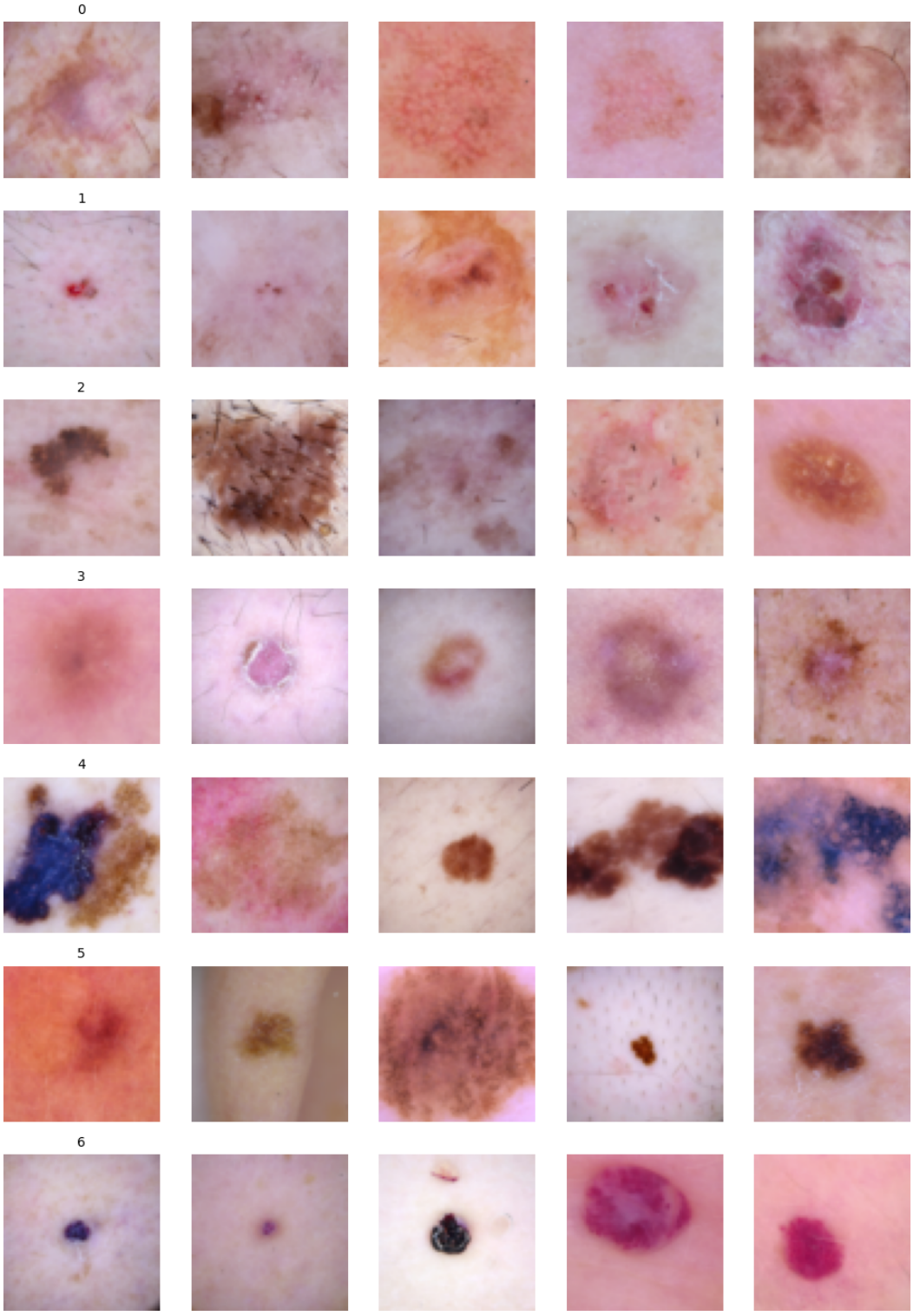

FIGURE 3.1 – Exemples d'images extraites du jeu `DermaMNIST`, illustrant les 7 types de lésions dermatologiques.

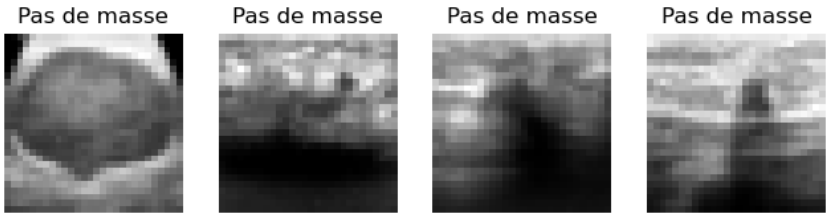

FIGURE 3.2 – Exemples d'images `BreastMNIST` sans masse tumorale.

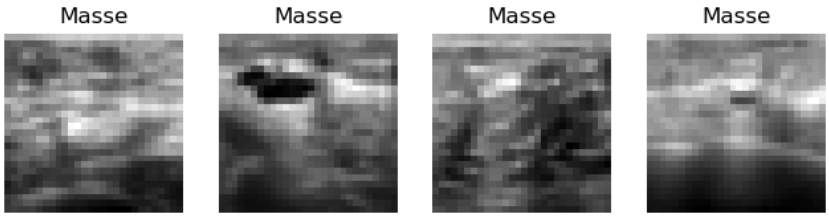

FIGURE 3.3 – Exemples d'images `BreastMNIST` avec masse tumorale.

3.2.3 Modélisation

Modélisation – Dataset DermaMNIST

Pour la classification des images dermatologiques issues du dataset *DermaMNIST*, nous avons conçu un modèle de réseau de neurones convolutionnel (*CNN*) simple et modulaire. Ce modèle, nommé `SimpleCNN`, vise à capturer les structures visuelles essentielles des images tout en restant suffisamment léger pour permettre des analyses explicatives ultérieures.

Architecture du modèle

Le modèle repose sur une succession de couches de convolution suivies de fonctions d'activation ReLU et de couches de *max pooling*. La complexité de l'architecture est configurable via deux hyperparamètres : le nombre de couches convolutionnelles (`num_conv_layers`) et le nombre initial de canaux (`base_channels`). Chaque bloc double la largeur des canaux.

Le squelette de la classe peut être résumé comme suit :

```
class SimpleCNN(nn.Module):
def __init__(self, num_classes=7, num_conv_layers=2,
   base_channels=32):
...
```

En sortie du bloc convolutionnel, les cartes de caractéristiques sont aplaties puis passées dans deux couches entièrement connectées avec régularisation par `dropout`. La couche finale contient 7 neurones, correspondant aux 7 classes de maladies.

Prétraitement des données

Les images du dataset ont été converties au format `ImageFolder`, puis prétraitées via les opérations suivantes :

— Conversion en niveaux de gris

— Redimensionnement à 64×64 pixels
— Normalisation via `ToTensor()`

Entraînement du modèle

L'entraînement a été réalisé sur GPU (si disponible), en utilisant l'optimiseur `Adam`, un taux d'apprentissage de 10^{-3}, une fonction de perte `CrossEntropyLoss` et un `batch size` de 64. Le processus a été mené sur 5 époques.

La boucle d'apprentissage principale peut être schématisée ainsi :

```
for epoch in range(num_epochs):
for images, labels in train_loader:
...
outputs = model(images)
loss = criterion(outputs, labels)
loss.backward()
optimizer.step()
```

Sauvegarde du modèle

À la fin de l'entraînement, les poids du modèle ont été sauvegardés à l'aide de la commande suivante :

```
torch.save(model.state_dict(), "saved_model/model.pt")
```

Ce fichier peut ensuite être rechargé pour des usages ultérieurs comme suit :

```
model = SimpleCNN(...)
model.load_state_dict(torch.load("saved_model/model.pt"))
model.eval()
```

Performances du modèle

L'évaluation sur l'ensemble de test a donné une précision finale de **68.08%**, témoignant d'une bonne capacité de généralisation du modèle sur ce problème de classification multi-classe.

```
Test Accuracy: 68.08%
```

Modélisation – Dataset BreastMNIST

Le dataset *BreastMNIST* est un jeu de données médicales issu de la base *MedMNIST*, contenant des images échographiques du sein annotées pour la classification binaire (lésion bénigne ou maligne). Dans cette section, nous présentons l'architecture utilisée ainsi que les étapes de prétraitement, d'entraînement et d'évaluation du modèle.

Architecture du modèle

Le modèle utilisé pour ce problème est un CNN simple, adapté à la classification binaire. Il est composé de blocs convolutionnels suivis de `ReLU` et de `MaxPooling`, puis de couches entièrement connectées avec un neurone de sortie (activation sigmoïde). L'architecture est la suivante :

— **Bloc 1** : Conv2D(1, 32, kernel=3, padding=1) → ReLU → MaxPool2D(2)
— **Bloc 2** : Conv2D(32, 64, kernel=3, padding=1) → ReLU → MaxPool2D(2)
— **Fully Connected** : Flatten → Linear($16 \times 16 \times 64$, 128) → ReLU → Dropout(0.5) → Linear(128, 1) → Sigmoid

Le début de la classe peut s'écrire ainsi :

```
class BreastCNN(nn.Module):
def __init__(self):
super().__init__()
...
```

Prétraitement des données

Chaque image a été convertie en niveaux de gris et redimensionnée à 64×64 pixels. Les transformations appliquées incluent :

— `Grayscale()`
— `Resize((64, 64))`
— `ToTensor()`

Le dataset a été divisé en trois sous-ensembles : entraînement, validation et test, en conservant les proportions initiales proposées par *MedMNIST*.

Entraînement du modèle

L'entraînement a été effectué sur 5 époques avec l'optimiseur `Adam`, un taux d'apprentissage de 10^{-3}, une fonction de perte `BCEWithLogitsLoss` (adaptée à la classification binaire) et un `batch size` de 64. La boucle d'entraînement est semblable à :

```
for epoch in range(num_epochs):
for images, labels in train_loader:
...
outputs = model(images).squeeze()
loss = criterion(outputs, labels.float())
loss.backward()
optimizer.step()
```

Sauvegarde et rechargement du modèle

Les poids du modèle ont été sauvegardés à l'aide de :

```
torch.save(model.state_dict(), "saved_model/breast_model.pt")
```

Puis rechargés pour l'évaluation :

```
model = BreastCNN()
model.load_state_dict(torch.load("saved_model/breast_model.pt"))
model.eval()
```

Performances

L'évaluation finale sur l'ensemble de test donne une précision d'environ **85.6%**, ce qui reflète la capacité du modèle à distinguer les deux types de lésions mammaires sur ce dataset.

```
Test Accuracy: 85.6%
```

3.2.4 Application des méthodes d'explicabilité sur DermaMNIST

Afin de comprendre les décisions prises par le modèle de classification convolutif entraîné sur le jeu de données DermaMNIST, nous avons appliqué trois techniques d'explicabilité visuelle : Integrated Gradients (IG), Saliency Maps et Grad-CAM. Ces méthodes permettent de générer des cartes d'importance qui mettent en évidence les régions de l'image les plus influentes dans la prédiction finale.

Préparation de l'image d'entrée

Nous avons sélectionné une image du jeu de test représentant une lésion cutanée en niveaux de gris. Cette image est d'abord convertie en tenseur PyTorch, puis transmise au modèle pour obtenir la classe prédite :

```
idx = 7
image, label = test_dataset[idx]
input_tensor = image.unsqueeze(0).to(device)
output = model(input_tensor)
pred_class = output.argmax(dim=1).item()
```

Méthode Integrated Gradients

La méthode des *Integrated Gradients* attribue une importance à chaque pixel en intégrant les gradients le long d'un chemin linéaire entre une image de référence (par exemple, noire) et l'image d'entrée réelle :

```
ig = IntegratedGradients(model)
attr_ig = ig.attribute(input_tensor, target=pred_class)
attr_ig = np.abs(attr_ig.squeeze().cpu().detach().numpy())
```

Méthode Saliency Maps

Les cartes de saillance (*Saliency Maps*) mesurent la sensibilité de la sortie du modèle par rapport à l'image d'entrée. Elles font apparaître les pixels dont une légère variation affecte fortement la prédiction :

```
saliency = Saliency(model)
attr_sal = saliency.attribute(input_tensor, target=pred_class)
attr_sal = np.abs(attr_sal.squeeze().cpu().detach().numpy())
```

Méthode Grad-CAM

La méthode Grad-CAM (*Gradient-weighted Class Activation Mapping*) repose sur les gradients de la dernière couche convolutive du modèle. Elle permet de localiser les zones importantes à l'aide d'une carte de chaleur :

```
for module in reversed(model.conv_layers):
if isinstance(module, nn.Conv2d):
last_conv = module
break

gradcam = LayerGradCam(model, last_conv)
attr_gc = gradcam.attribute(input_tensor, target=pred_class)
attr_gc = F.interpolate(attr_gc, size=(64, 64), mode='bilinear')
attr_gc = np.maximum(attr_gc.squeeze().cpu().detach().numpy(), 0)
```

Affichage des visualisations

Les différentes visualisations générées sont ensuite affichées côte à côte pour permettre une comparaison directe entre les méthodes :

```
fig, axs = plt.subplots(1, 4, figsize=(12, 4))
axs[0].imshow(image_np, cmap='gray'); axs[0].set_title("Image
    originale")
axs[1].imshow(attr_ig, cmap='inferno'); axs[1].set_title("
    Integrated Gradients")
axs[2].imshow(attr_sal, cmap='inferno'); axs[2].set_title("
    Saliency")
axs[3].imshow(attr_gc, cmap='inferno'); axs[3].set_title("Grad-
    CAM")
```

Résultats obtenus

Nous avons ainsi pu générer une image comparative illustrant les résultats des trois méthodes d'explicabilité appliquées à une même lésion cutanée. Cette image est présentée à la figure 3.4.

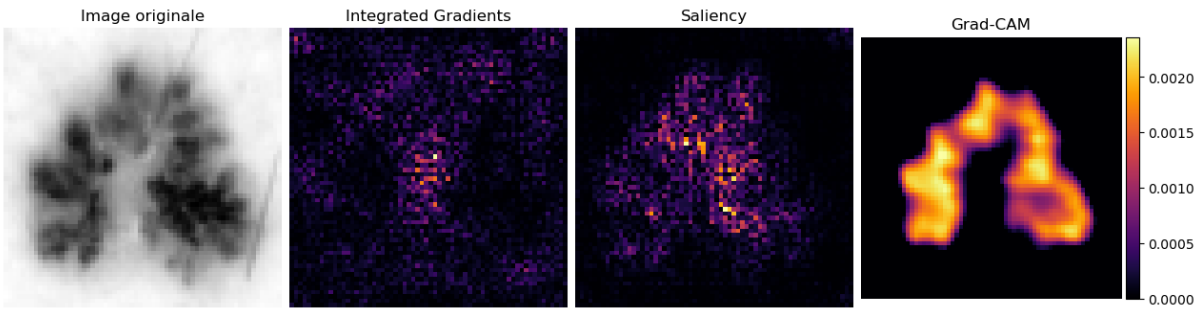

FIGURE 3.4 – Visualisation des méthodes d'explicabilité (Integrated Gradients, Saliency Maps, Grad-CAM) appliquées à une image du dataset DermaMNIST.

Application des méthodes d'explicabilité sur BreastMNIST

Afin de mieux comprendre les décisions du modèle CNN entraîné sur le jeu de données *BreastMNIST*, nous avons appliqué trois méthodes d'explicabilité visuelle : Integrated Gradients, Saliency Maps, et Grad-CAM. Une image test a été sélectionnée aléatoirement

pour illustrer l'analyse des régions discriminantes qui influencent la classification binaire (présence ou non de masse mammaire).

— Le modèle est mis en mode évaluation grâce à `model.eval()`.

— Une image test est extraite, transformée en tenseur, puis passée au modèle pour obtenir une prédiction.

— Trois méthodes d'explicabilité ont été appliquées :

1. **Integrated Gradients (IG)** : basée sur le chemin intégré entre une base neutre (baseline) et l'image d'entrée.
2. **Saliency Maps** : évalue la sensibilité de la prédiction par rapport à chaque pixel.
3. **Grad-CAM** : exploite les gradients de la dernière couche convolutive pour générer une carte d'activation spatiale.

— Les résultats ont été affichés sous forme de figures côte à côte, accompagnées de barres de couleur pour faciliter la lecture des intensités.

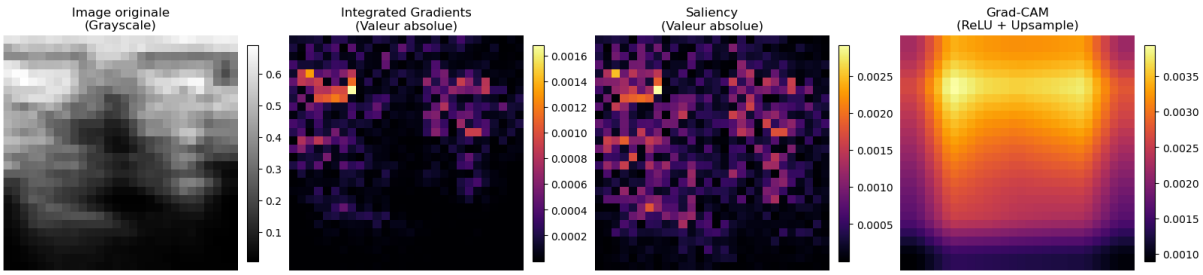

FIGURE 3.5 – Visualisation de l'image test (BreastMNIST) et des cartes d'explicabilité : Integrated Gradients, Saliency, Grad-CAM

3.2.5 Analyse et interprétation des visualisations explicatives

Comparaison qualitative des méthodes d'explicabilité

Les trois méthodes visuelles utilisées – *Integrated Gradients*, *Saliency Maps* et *Grad-CAM* – ont permis de générer des cartes d'importance mettant en évidence les régions des images médicales (DermaMNIST et BreastMNIST) ayant le plus influencé la prédiction du modèle.

D'un point de vue qualitatif, les trois approches convergent globalement vers des zones similaires, souvent centrées sur la région pathologique, tout en présentant des nuances notables :

— **Integrated Gradients (IG)** produit des cartes fines, sensibles aux gradients internes du modèle. Les activations sont souvent précises et concentrées au centre de la lésion (DermaMNIST) ou de la masse (BreastMNIST), avec des dégradés d'intensité mettant en valeur la texture.

— **Saliency Maps** génère des cartes plus brutes, parfois bruitées, mais efficaces pour détecter les zones à forte sensibilité locale. Cette méthode a l'avantage de refléter directement les variations les plus critiques sur les pixels.

— **Grad-CAM**, en se basant sur les activations des couches profondes, offre une vue plus globale. Les régions mises en évidence sont plus diffuses, mais souvent bien centrées sur la lésion ou la masse suspecte. Cela rend cette méthode particulièrement lisible, au prix d'une résolution spatiale plus faible.

Cohérence avec les objectifs cliniques

Les régions activées par les modèles à travers les différentes méthodes d'explicabilité correspondent généralement à des zones cliniquement pertinentes. Sur les images dermatologiques, les activations se concentrent sur la partie centrale des lésions, là où se manifestent des signes typiques de pathologies cutanées (asymétrie, irrégularité des bords, variation de texture ou de couleur).

De manière similaire, les mammographies du dataset BreastMNIST montrent que les activations se focalisent sur les masses suspectes, en soulignant à la fois leur centre et leurs contours irréguliers. Ces observations rejoignent les critères utilisés en radiologie pour l'analyse des tumeurs mammaires : asymétrie, densité, hétérogénéité, bords flous ou mal définis.

Cette convergence entre les zones activées par le modèle et les régions d'intérêt clinique suggère que le réseau a appris à extraire des motifs visuellement significatifs, similaires à ceux utilisés par les professionnels de santé.

Lisibilité pour un utilisateur non spécialiste

Un aspect crucial de l'explicabilité est la capacité des méthodes à être comprises par des utilisateurs non experts, comme les médecins ou les patients. À ce titre :

- **Grad-CAM** apparaît comme la méthode la plus accessible. Les cartes de chaleur générées sont intuitives et permettent de visualiser rapidement les zones importantes sans connaissances techniques avancées.
- **Integrated Gradients** fournit des résultats plus subtils, qui peuvent nécessiter une explication ou une mise en contexte pour être interprétés correctement par un clinicien.
- **Saliency Maps**, en raison de leur aspect parfois bruité, sont les moins lisibles pour un public non technique, bien qu'elles soient informatives sur le fonctionnement du modèle.

Ainsi, dans une perspective d'intégration dans un outil d'aide au diagnostic, Grad-CAM constitue une méthode particulièrement intéressante, car elle combine lisibilité, pertinence visuelle, et interprétabilité partielle des décisions du modèle.

3.3 Etude sur les données médicales non visuelles

Contrairement aux données visuelles, les données médicales tabulaires représentent des mesures cliniques structurées. Nous étudions ici leur traitement, modélisation et interprétation à l'aide de techniques d'explicabilité adaptées

3.3.1 Présentation des jeux de données

Dans cette étude, trois jeux de données médicaux non visuels ont été utilisés afin d'évaluer les performances des modèles de classification et d'appliquer des techniques d'explicabilité sur des données tabulaires. Ces jeux de données, largement répandus dans la littérature scientifique, sont issus de contextes cliniques réels et comportent exclusivement des variables numériques ou discrètes.

Breast Cancer Wisconsin (Diagnostic) Ce jeu de données provient de la *University of Wisconsin Hospital* et concerne le diagnostic de tumeurs mammaires à partir de mesures numériques extraites d'images de cytologie. Chaque observation correspond à une lésion observée via aspiration à l'aiguille fine, caractérisée par des descripteurs morphologiques calculés sur l'image.[27]

- **Objectif** : classification binaire (tumeur bénigne vs maligne).
- **Nombre d'observations** : 569.
- **Nombre de variables** : 30 caractéristiques numériques continues (rayon, texture, concavité, symétrie, etc.).
- **Variable cible** : type de tumeur (0 = bénigne, 1 = maligne).
- **Type de données** : numériques tabulaires, normalisées.

Pima Indians Diabetes Ce dataset est fourni par le *National Institute of Diabetes and Digestive and Kidney Diseases* et regroupe des données cliniques relatives à des patientes amérindiennes d'origine Pima. Il a pour but de prédire l'apparition du diabète à partir de paramètres physiologiques mesurés lors de consultations.[26]

- **Objectif** : classification binaire (présence ou absence de diabète).
- **Nombre d'observations** : 768.
- **Nombre de variables** : 8 variables cliniques numériques (glycémie, pression artérielle, IMC, etc.).
- **Variable cible** : diagnostic de diabète (0 = non diabétique, 1 = diabétique).
- **Type de données** : numériques tabulaires (sans variables catégorielles).

Heart Disease (Cleveland) Ce jeu de données est extrait de l'étude cardiovasculaire menée par la *Cleveland Clinic Foundation*. Il vise à prédire la présence d'une maladie cardiaque chez des patients à partir de paramètres cliniques standards et de résultats d'examens complémentaires.[10]

- **Objectif** : classification binaire (absence ou présence de maladie cardiaque).
- **Nombre d'observations** : 303.
- **Nombre de variables** : 13 caractéristiques (âge, sexe, cholestérol, électrocardiogramme, etc.).
- **Variable cible** : présence de maladie (0 = absence, 1 = présence).
- **Type de données** : mixtes (numériques continues et catégorielles encodées en entiers).

3.3.2 Prétraitement et exploration de données

Les différentes étapes de préparation et d'analyse exploratoire ont été appliquées aux trois jeux de données sélectionnés. Étant donné que leur description détaillée a été présentée dans la section précédente, cette partie se focalise exclusivement sur le traitement des données et les visualisations utilisées pour mieux comprendre leur structure.

Breast Cancer Wisconsin

Après séparation en ensembles d'entraînement et de test (80/20), les variables explicatives ont été normalisées à l'aide d'un *StandardScaler*. Une Analyse en Composantes Principales (ACP) a été réalisée afin de visualiser la structure des données dans un espace

bidimensionnel. Cette représentation révèle une certaine séparation entre les échantillons bénins et malins.

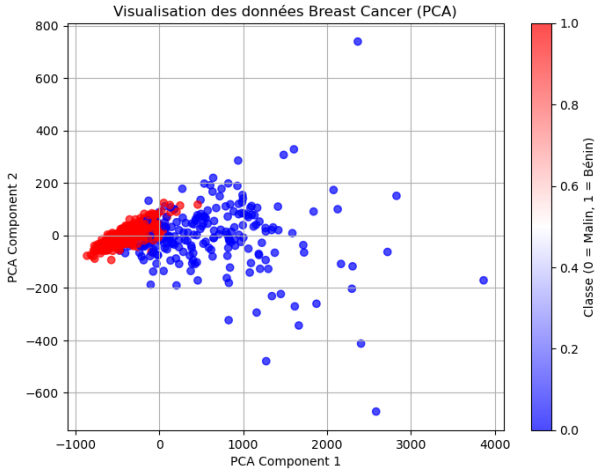

FIGURE 3.6 – Projection en 2D par ACP – Breast Cancer Wisconsin

Cleveland Heart Disease

Les valeurs manquantes (indiquées par " ?") ont été remplacées par des valeurs manquantes explicites, puis imputées par la médiane. Les types ont été convertis en format numérique. La variable cible a été binarisée (0 : absence de maladie, 1 : présence).

L'exploration statistique s'est appuyée sur une matrice de corrélation et des boxplots, permettant d'identifier des relations significatives entre certaines variables cliniques (âge, cholestérol, fréquence cardiaque maximale, etc.) et la cible.

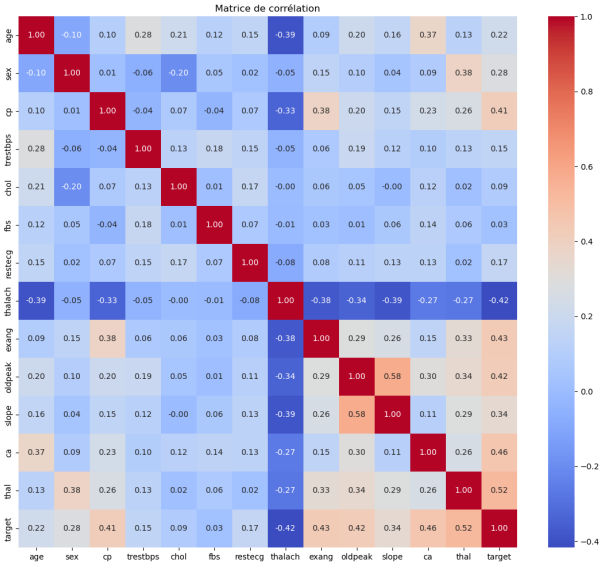

FIGURE 3.7 – Matrice de corrélation – Cleveland Heart Disease

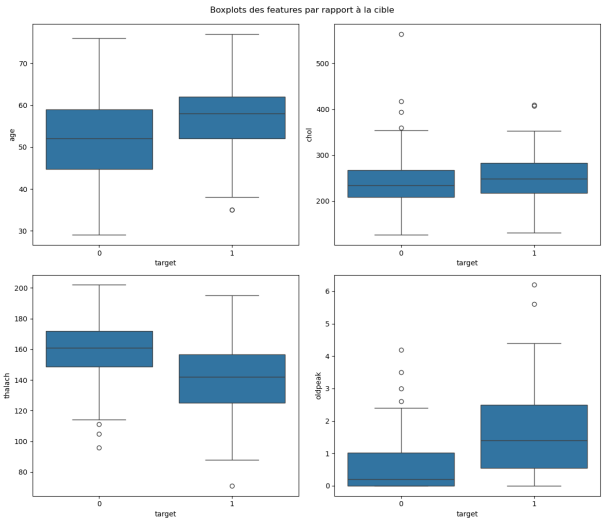

FIGURE 3.8 – Boxplots des variables selon la présence de maladie

Pima Indians Diabetes

Les valeurs nulles implicites (valeurs à zéro biologiquement impossibles) ont été identifiées et remplacées lorsque pertinent. Les données ont ensuite été normalisées après division en ensembles d'apprentissage et de test.

Une analyse visuelle a été réalisée à travers des histogrammes, des boxplots univariés et bivariés, ainsi qu'une matrice de corrélation. Ces visualisations ont mis en évidence des variables particulièrement discriminantes comme la glycémie (glucose), l'IMC et l'âge.

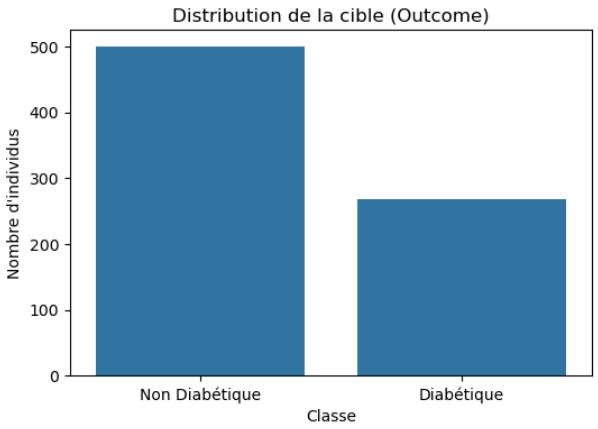

FIGURE 3.9 – Distribution de la variable cible (Outcome) – Pima

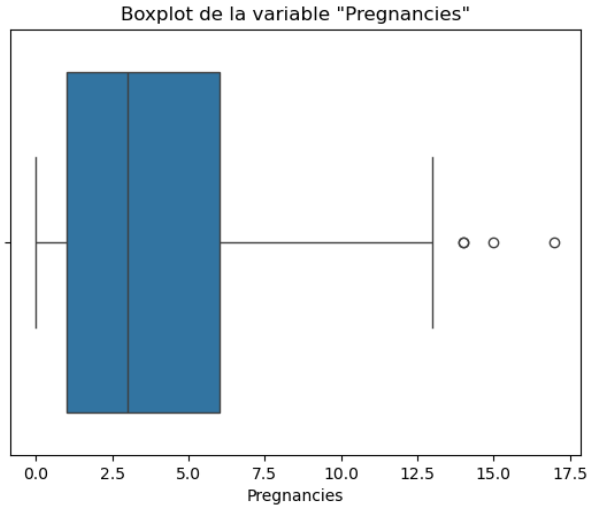

FIGURE 3.10 – Boxplots des variables selon le statut diabétique

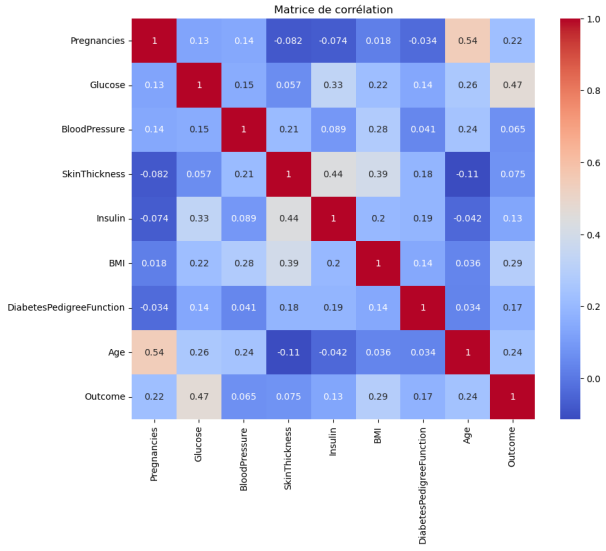

FIGURE 3.11 – Matrice de corrélation – Pima Indians Diabetes

3.3.3 Modélisation

Cette section détaille le processus de modélisation appliqué aux trois jeux de données étudiés. Pour chacun, un réseau de neurones a été entraîné avec une architecture adaptée à la nature des variables et à la tâche de classification binaire. Les modèles ont ensuite été évalués à l'aide de métriques classiques, en vue d'une interprétation ultérieure par des méthodes d'explicabilité.

Pima Indians Diabetes

Le premier modèle a été entraîné sur les données cliniques des patientes atteintes ou non de diabète de type 2. Un perceptron multicouche (MLP) a été utilisé avec deux couches cachées de tailles (64, 32), une fonction d'activation `ReLU` et l'optimiseur `Adam`. Les données ont été normalisées au préalable à l'aide du `StandardScaler` :

Listing 3.1 – Normalisation des données

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Le modèle a ensuite été entraîné avec arrêt anticipé pour prévenir le surapprentissage :

Listing 3.2 – Définition du MLP

```
mlp = MLPClassifier(hidden_layer_sizes=(64, 32),
activation='relu',
solver='adam',
max_iter=1000,
early_stopping=True,
random_state=42)
mlp.fit(X_train_scaled, y_train)
```

L'évaluation du modèle repose sur plusieurs métriques : la précision globale, la matrice de confusion et la courbe ROC. Le modèle atteint une précision satisfaisante sur les données de test. La courbe ROC a révélé une capacité discriminante notable.

Cleveland Heart Disease

Pour le jeu de données Cleveland, un MLP plus profond a été mis en place afin de mieux capturer les interactions non linéaires entre variables. L'architecture est composée de trois couches cachées successives : 128, 64 et 32 neurones. La procédure de normalisation et de séparation train/test est identique à celle utilisée précédemment.

Listing 3.3 – Architecture du MLP pour Cleveland

```
mlp = MLPClassifier(hidden_layer_sizes=(128, 64, 32),
activation='relu',
solver='adam',
max_iter=1000,
early_stopping=True,
random_state=42)
mlp.fit(X_train_scaled, y_train)
```

L'évaluation a mis en évidence une bonne performance sur la prédiction de la présence de maladie cardiaque. Une matrice de confusion et un rapport de classification ont été générés pour caractériser les performances par classe.

Breast Cancer Wisconsin

Dans ce cas, un modèle de réseau de neurones a été implémenté via la bibliothèque `TensorFlow / Keras`, afin d'avoir un meilleur contrôle sur l'architecture et de rendre le modèle compatible avec certaines méthodes d'explicabilité par backpropagation. L'architecture est restée simple : deux couches cachées suivies d'une sortie linéaire.

Listing 3.4 – Architecture du réseau Keras

```
model = Sequential([
Dense(64, activation='relu', input_shape=(X_train_scaled.shape
    [1],)),
Dense(32, activation='relu'),
Dense(1, activation='linear')

])
model.compile(optimizer='adam', loss='mse')
model.fit(X_train_scaled, y_train, epochs=10, batch_size=32)
```

Même si la tâche est de classification, une fonction de coût quadratique (`MSE`) a été utilisée ici à dessein, afin de faciliter l'application ultérieure de techniques explicatives basées sur les gradients.

L'évaluation a permis de confirmer une bonne séparation entre les classes. La matrice de confusion générée montre une performance robuste, avec peu d'erreurs sur les données de test.

Tous les modèles ont été sauvegardés à l'aide de la bibliothèque `joblib` afin de permettre leur réutilisation dans les phases d'explicabilité à venir.

3.3.4 Application d'explicabilité

Cette section présente l'application concrète de plusieurs méthodes d'explicabilité sur les modèles entraînés à partir de données médicales tabulaires. Pour chaque jeu de données étudié, nous sélectionnons un ensemble de techniques adaptées, en alternant approches locales et globales.

Notre démarche est la suivante : nous préparons les données nécessaires à l'explication (modèle entraîné, jeu de test, prédictions), puis nous appliquons chaque méthode en mettant en évidence les portions pertinentes de code, les visualisations produites et l'interprétation des résultats.

Nous commençons cette étude par le jeu de données Pima Indians Diabetes.

Méthode SHAP : Application au modèle MLP sur le dataset Pima Indians Diabetes

Préparation des données et mise en place de l'explainer Afin d'interpréter les prédictions du modèle MLP entraîné sur le jeu de données Pima Indians Diabetes, nous avons recours à l'approche SHAP (SHapley Additive exPlanations), qui repose sur la théorie des valeurs de Shapley. Étant donné la nature "boîte noire" du réseau de neurones, nous utilisons le KernelExplainer, compatible avec les modèles non interprétables de manière intrinsèque.

Extrait de code :

```
import shap
background_data = shap.kmeans(X_train_scaled, 10)
explainer_shap = shap.KernelExplainer(mlp.predict_proba,
   background_data)
shap_values = explainer_shap.shap_values(X_test_scaled[:100])
```

Nous sélectionnons un sous-échantillon représentatif du jeu de données (via KMeans) pour servir de distribution de référence (background). L'explicateur est ensuite appliqué à 100 instances du jeu de test afin d'évaluer l'impact de chaque feature sur les prédictions de probabilité de la classe positive (présence de diabète).

Visualisation globale : Summary Plot Le summary plot SHAP permet de synthétiser l'influence de chaque variable sur les sorties du modèle. Chaque point représente une instance du jeu de test ; sa position horizontale indique l'impact de la variable sur la prédiction, et sa couleur correspond à la valeur réelle de cette variable (du bleu pour les valeurs faibles au rouge pour les valeurs élevées).

Extrait de code :

```
shap.summary_plot(
shap_values=shap_values_class1,
features=X_test_global,
feature_names=columns[:-1],
plot_type="dot"
)
```

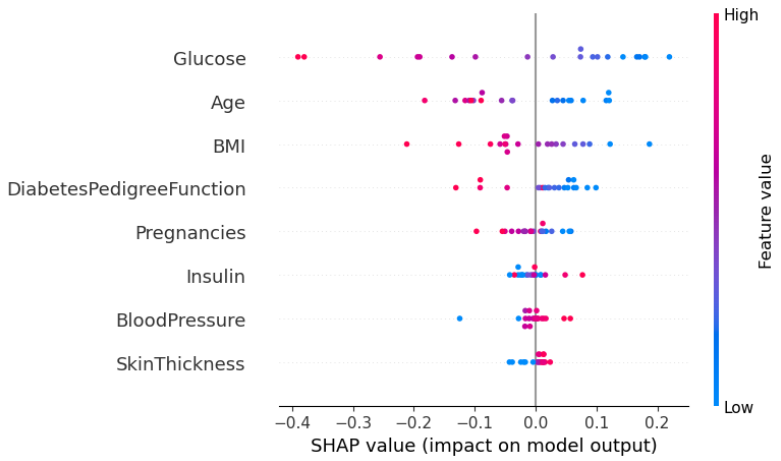

FIGURE 3.12 – Summary plot des valeurs SHAP sur le dataset Pima Indians Diabetes

Analyse et interprétation Cette visualisation met en évidence les variables les plus déterminantes dans les prédictions du modèle. On observe notamment :

Le glucose est la variable la plus influente : des valeurs élevées (en rouge) contribuent fortement à augmenter la probabilité de diabète (points rouges à droite de l'axe Y = 0).

L'IMC (BMI) et le nombre de grossesses (Pregnancies) ont également un impact significatif : des valeurs élevées de ces variables augmentent la prédiction positive.

D'autres variables, comme la pression artérielle ou l'épaisseur de la peau, ont un impact beaucoup plus modéré, leurs points étant concentrés autour de zéro.

L'âge et la fonction génétique (DiabetesPedigreeFunction) ont un effet intermédiaire.

Ce graphe offre une vue d'ensemble précieuse sur l'importance des variables et la direction de leur effet sur les sorties du modèle.

Interprétation locale avec `shap.force_plot` Pour expliquer une prédiction individuelle du modèle MLP, nous utilisons le `force_plot` de SHAP, qui visualise l'impact de chaque variable sur la sortie finale.

Code d'affichage :

```
shap.force_plot(
    explainer.expected_value[1],
    shap_values[1][0],
    features=X_test.iloc[0],
    feature_names=X_test.columns
)
```

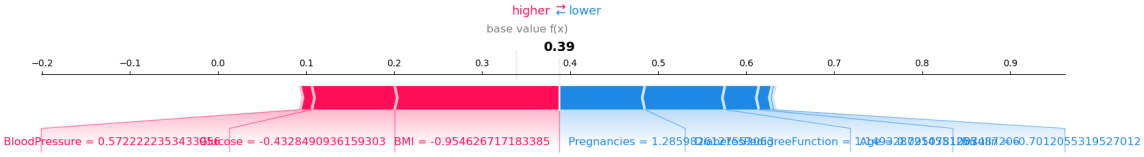

FIGURE 3.13 – Force plot pour une instance du jeu Pima Indians Diabetes

La ligne centrale représente la prédiction moyenne du modèle (*base value* $\approx 0{,}39$). Les variables qui augmentent la probabilité de diabète apparaissent en rouge, celles qui la diminuent en bleu.

Analyse :

- **Glucose** diminue fortement la probabilité (valeur SHAP négative importante).
- **Pregnancies** et **BMI** augmentent fortement la prédiction.
- **BloodPressure** et **DiabetesPedigreeFunction** ont un effet modéré.

Ce graphique met en évidence les facteurs principaux ayant conduit à une prédiction élevée pour cette instance.

Interprétation locale avec `shap.waterfall_plot` Le `waterfall_plot` permet de visualiser l'impact cumulatif de chaque variable sur la prédiction finale pour une instance spécifique.

Code d'affichage :

```
from shap import Explanation
exp = Explanation(
    values=shap_values_class1[instance_idx],
    base_values=explainer_shap.expected_value[1],
    data=X_test_small[instance_idx],
```

```
    feature_names=columns[:-1]
)
shap.waterfall_plot(exp)
```

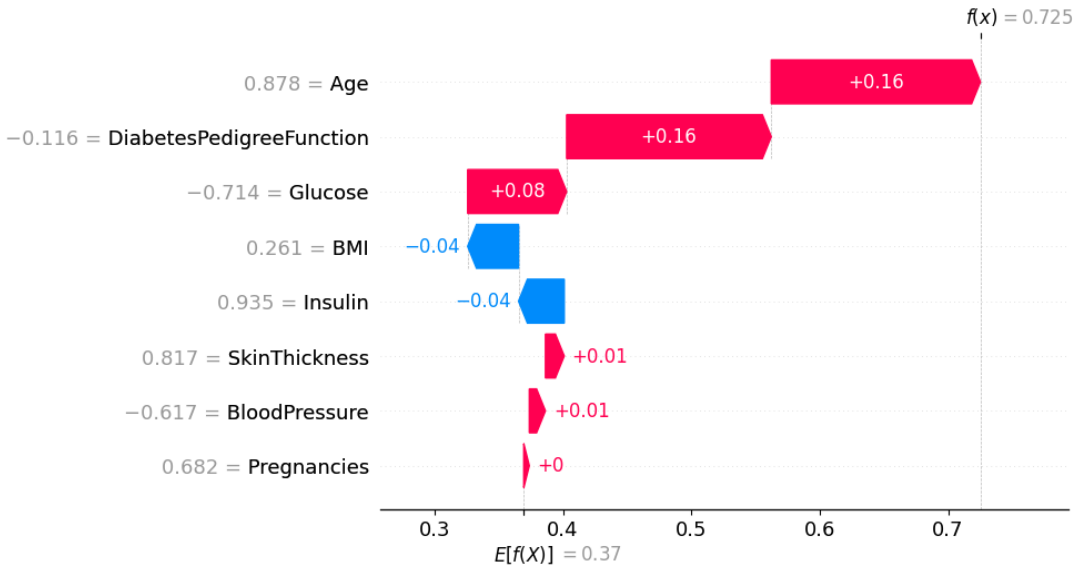

FIGURE 3.14 – Waterfall plot pour une instance du jeu Pima Indians Diabetes

La prédiction est obtenue en partant de la *base value* ($\approx 0{,}37$), puis en ajoutant les effets des variables :

- **Age** et **DiabetesPedigreeFunction** augmentent fortement la prédiction (barres rouges longues).
- **Glucose** a un effet modéré positif.
- **BMI** et **Insulin** réduisent légèrement la probabilité (barres bleues).
- **Pregnancies**, **SkinThickness** et **BloodPressure** ont un effet marginal.

Ce graphique permet d'identifier les principales contributions positives (âge, hérédité) et négatives (IMC, insuline) à la prédiction finale.

Importance globale des variables – `shap.summary_plot (bar)` Le `summary_plot` avec l'option `plot_type="bar"` permet d'identifier les variables les plus influentes sur l'ensemble du jeu de test.

Code d'affichage :

```
shap.summary_plot(
    shap_values_class1,
    X_test_small,
    feature_names=columns[:-1],
    plot_type="bar"
)
```

Chaque barre indique l'impact moyen absolu d'une variable sur la sortie du modèle (valeurs de SHAP moyennes en valeur absolue).

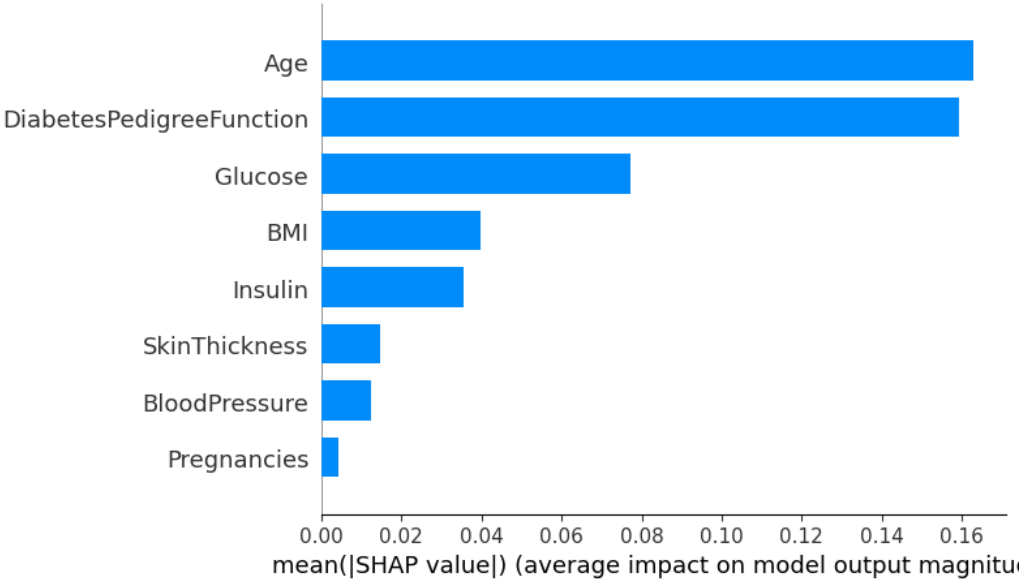

FIGURE 3.15 – Importance moyenne des variables selon SHAP

Les variables les plus importantes sont :

- **Age** : plus forte contribution moyenne.
- **DiabetesPedigreeFunction** et **Glucose** : effets également marqués.
- **BMI**, **Insulin**, **SkinThickness**, **BloodPressure**, **Pregnancies** : contributions plus faibles.

Ce graphique permet de détecter les variables dominantes (âge, hérédité, glucose) ainsi que celles à impact mineur, pouvant être écartées pour un modèle plus simple.

Relation entre Glucose et la prédiction – `shap.dependence_plot` Pour étudier l'influence directe de la variable `Glucose` sur les prédictions du modèle, nous utilisons le `dependence_plot` de SHAP.

Code d'affichage :

```
shap.dependence_plot(
    "Glucose",
    shap_values_class1,
    X_test_global,
    feature_names=columns[:-1]
)
```

Ce graphique met en évidence l'impact de la variable `Glucose` sur la sortie du modèle, ainsi que son éventuelle interaction avec une autre variable (`BloodPressure`, ici utilisée pour la coloration des points).

Analyse du graphique :

- **Axe horizontal** : valeurs de `Glucose` (probablement standardisées).
- **Axe vertical** : valeurs SHAP associées à `Glucose`, indiquant leur contribution à la prédiction (positive ou négative).
- **Couleur des points** : valeurs de `BloodPressure`, de bleu (faible) à rouge (élevé).

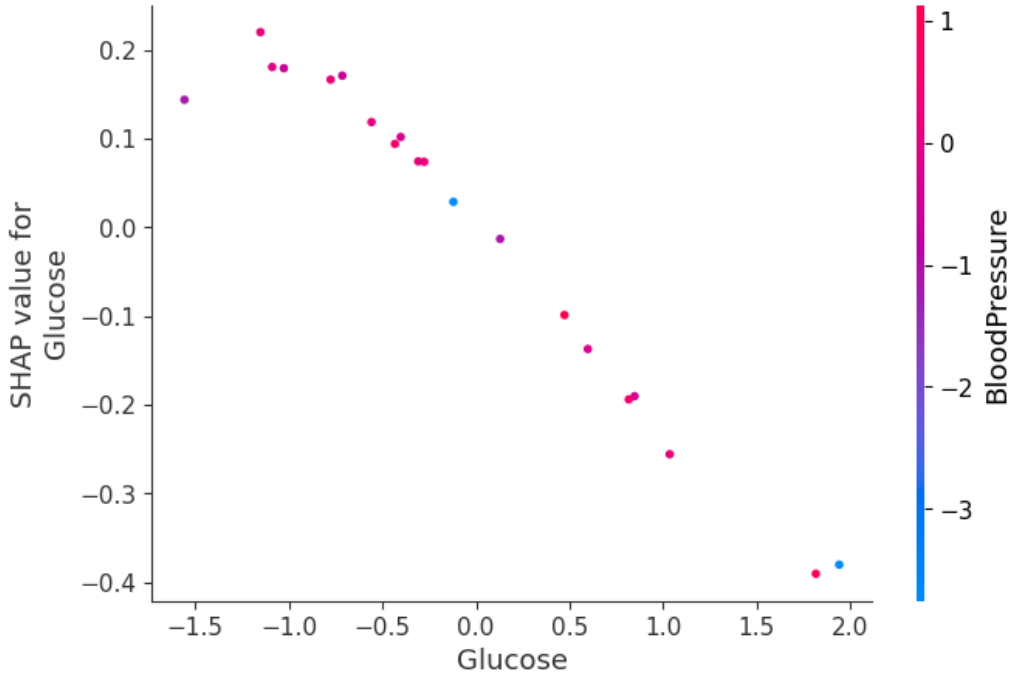

FIGURE 3.16 – Relation entre les valeurs de Glucose et leur impact (valeurs SHAP)

Observations clés :

— Une relation globalement croissante est observée : plus la valeur de `Glucose` est élevée, plus sa contribution à la probabilité de diabète (valeurs SHAP positives) est importante, ce qui est cohérent avec les connaissances médicales.

— Les points sont relativement dispersés, indiquant une variabilité interindividuelle : l'influence de `Glucose` dépend potentiellement d'autres variables.

— Aucune interaction marquée n'est visible avec `BloodPressure` : les couleurs sont réparties de manière aléatoire, sans tendance apparente.

Utilité du graphique : Ce type de visualisation est précieux pour :

— Comprendre comment une variable spécifique influence la prédiction.

— Détecter d'éventuelles interactions avec d'autres variables.

— Identifier des comportements contre-intuitifs ou suspects.

Bar plot local : interprétation de la prédiction d'une instance spécifique Pour mieux comprendre la décision du modèle pour une instance donnée, nous utilisons le graphique de type `decision_plot`, qui illustre la contribution de chaque variable à la prédiction finale.

Code utilisé :

```
shap.decision_plot(
    explainer_shap.expected_value[1],
    shap_values_class1[instance_idx],
    X_test_small[instance_idx],
    feature_names=columns[:-1],
    show=False
)
plt.show()
```

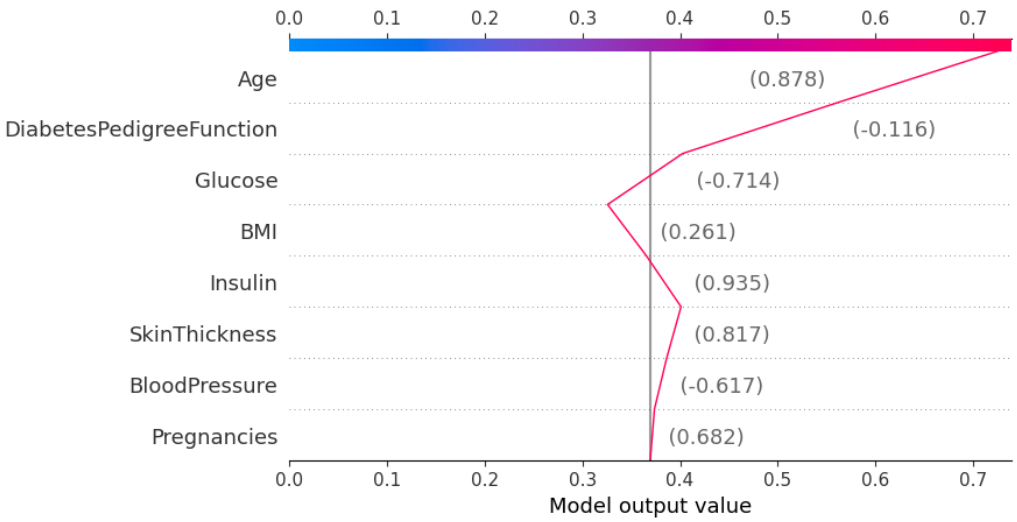

FIGURE 3.17 – Bar plot local : contribution des features pour une instance spécifique

Structure du graphique :

- **Axe horizontal** : valeur de sortie du modèle (probabilité de diabète).
- **Base value** : environ 0,4, correspond à la prédiction moyenne du modèle.
- **Barres colorées** : représentent les contributions individuelles des variables :
  - Bleu : contribution positive (augmentation de la probabilité de diabète).
  - Rouge : contribution négative (diminution de la probabilité de diabète).
- Chaque barre est annotée avec la valeur réelle de la feature pour cette instance.

Analyse de la prédiction :

- **Variables à impact positif** :
  - **Age** (valeur : 0.878) : exerce l'effet le plus fort en faveur de la prédiction de diabète.
  - **Pregnancies** et **BMI** : contribuent également positivement, avec un impact modéré.
- **Variables à impact négatif** :
  - **Glucose** (valeur : -0.714) : diminue fortement la probabilité de diabète pour cette instance.
  - **DiabetesPedigreeFunction**, **BloodPressure**, **Insulin**, **SkinThickness** : diminuent légèrement la prédiction, avec un impact modéré à faible.
- **Prédiction finale** :
  - En combinant tous les effets, la sortie du modèle se rapproche de 0.7, indiquant une probabilité élevée de diabète pour cette instance.
  - Ce résultat s'explique principalement par l'âge avancé de l'individu, qui compense l'effet protecteur d'un taux de glucose relativement bas.

Intérêt du bar plot local :

- Permet de justifier la prédiction du modèle de façon transparente.
- Aide à identifier les variables déterminantes dans le cas d'une observation particulière.
- Peut alerter sur des déséquilibres ou des effets inattendus (ici, un glucose bas diminue la probabilité).

Explication locale avec LIME pour une instance spécifique

En complément des méthodes SHAP, nous avons appliqué LIME (*Local Interpretable Model-agnostic Explanations*) pour interpréter localement la décision du modèle MLP sur une instance spécifique du jeu de test. Cette méthode consiste à approximer localement le comportement du modèle par un classifieur linéaire simple, entraîné sur des échantillons perturbés autour de l'instance à expliquer.

Code utilisé :

```
from lime import lime_tabular

lime_explainer = lime_tabular.LimeTabularExplainer(
    training_data=X_train_scaled,
    feature_names=columns[:-1],
   class_names=['Non Diabetique', 'Diabetique'],

    mode='classification'
)

instance = X_test_scaled[0]
lime_exp = lime_explainer.explain_instance(instance, mlp.
   predict_proba, num_features=10)

html_lime = lime_exp.as_html()
with open("lime_local_explanation.html", "w", encoding="utf-8")
   as file:
    file.write(html_lime)
```

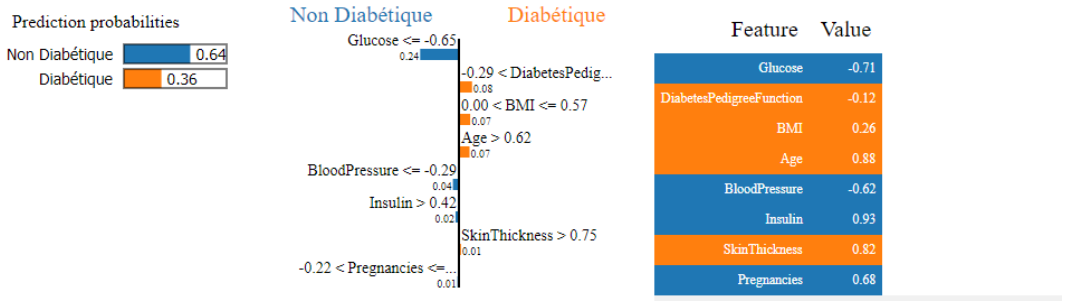

FIGURE 3.18 – Explication LIME locale pour une instance du jeu `Pima Indians Diabetes`

Analyse de la visualisation LIME :

- **Probabilités de prédiction** : Le modèle attribue une probabilité de 64% à la classe *Non Diabétique*, contre 36% pour la classe *Diabétique*.
- **Variables explicatives locales** : La figure met en évidence les variables ayant le plus influencé cette prédiction. Les contributions négatives (en bleu) tirent la prédiction vers la classe *Non Diabétique*, tandis que les contributions positives (en orange) favorisent la classe *Diabétique*.
  - **Glucose** (−0,71) et **BloodPressure** (−0,62) : facteurs protecteurs dominants.
  - **Insulin** (0,93) et **Age** (0,88) : principaux contributeurs au risque de diabète.

— **SkinThickness** (0,82) et **Pregnancies** (0,68) : effets positifs modérés.
— **Règles de décision locales** : LIME génère également des règles simples exprimant des intervalles sur les variables normalisées, permettant de comprendre le raisonnement local :
— `Glucose` $\leq -0{,}65$ et `BloodPressure` $\leq -0{,}29$: diminuent la probabilité.
— `Insulin` $> 0{,}42$ et `Age` $> 0{,}62$: augmentent la probabilité.

Conclusion :
— Cette explication locale éclaire le rôle précis de chaque variable dans la décision du modèle, à l'échelle individuelle.
— LIME permet ici de justifier une prédiction *Non Diabétique*, essentiellement influencée par un taux de glucose et une pression artérielle relativement bas, malgré la présence d'indicateurs de risque.
— Cette interprétation est précieuse pour l'audit du modèle et la validation clinique de ses décisions.

Importance des variables par permutation

Pour compléter l'analyse explicative du modèle, nous avons recours à la méthode d'importance par permutation (permutation importance). Cette approche consiste à mesurer l'impact de chaque variable sur la performance globale du modèle en évaluant la dégradation de sa précision lorsque les valeurs d'une variable donnée sont aléatoirement permutées. Plus la performance diminue, plus la variable est jugée importante.

Dans notre cas, cette méthode permet de déterminer quelles caractéristiques influencent le plus les prédictions du modèle de réseau de neurones entraîné sur le jeu de données Pima Indians Diabetes.

Implémentation en Python

Listing 3.5 – Calcul de l'importance par permutation

```
from sklearn.inspection import permutation_importance
import matplotlib.pyplot as plt

result = permutation_importance(mlp, X_test_scaled, y_test,
   n_repeats=10, random_state=42)

plt.figure(figsize=(8, 6))
plt.barh(columns[:-1], result.importances_mean)
plt.title("Feature Importance via Permutation")
plt.xlabel("Importance")
plt.ylabel("Features")
plt.tight_layout()
plt.show()
```

Le code ci-dessus utilise la fonction permutation importance de la bibliothèque `scikit-learn` pour estimer l'importance de chaque variable en perturbant son contenu. La représentation graphique est générée sous forme de diagramme en barres horizontales pour une lecture intuitive.

Visualisation des résultats

Analyse de la figure

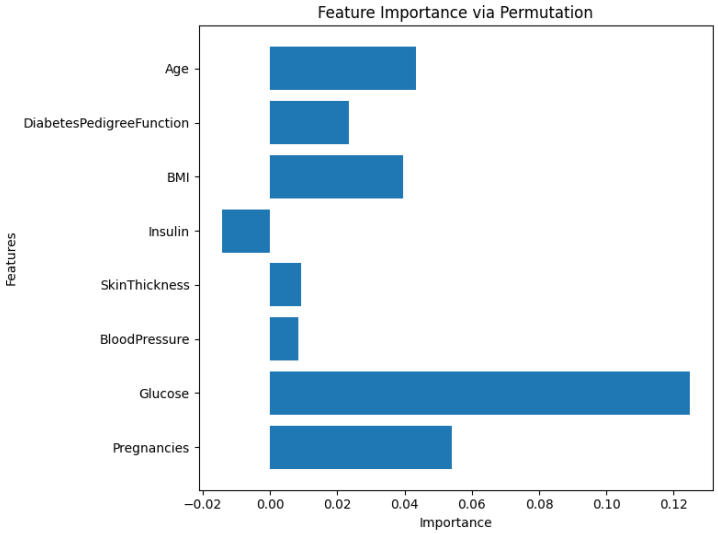

FIGURE 3.19 – Importance globale des variables selon la méthode de permutation

La figure 3.19 présente l'impact relatif de chaque variable sur la performance du modèle. Les résultats indiquent que :

- **Glucose** est de loin la variable la plus influente. Sa permutation entraîne une chute marquée des performances, ce qui confirme son rôle central dans la détection du diabète.
- **Pregnancies**, **Age** et **BMI** suivent, avec une influence modérée mais notable. Cela témoigne de leur contribution significative aux décisions du modèle.
- Les variables **DiabetesPedigreeFunction**, **Insulin**, **SkinThickness** et **BloodPressure** présentent une importance nettement plus faible. Leur permutation n'entraîne qu'une faible variation des performances.

Ces résultats complètent utilement les interprétations locales précédentes (SHAP et LIME) en offrant une perspective globale sur l'influence moyenne de chaque variable à l'échelle du jeu de test. La forte importance du glucose est cohérente avec ce que l'on observe dans la littérature médicale, renforçant la confiance dans le comportement du modèle.

Analyse de l'importance des variables avec ELI5

Nous avons utilisé la méthode de permutation via la bibliothèque `eli5` pour estimer l'influence globale de chaque variable d'entrée sur les prédictions du réseau MLP.

Code Python utilisé :

```
import eli5
from eli5.sklearn import PermutationImportance

X_test_df = pd.DataFrame(X_test_scaled, columns=columns[:-1])
perm = PermutationImportance(mlp, random_state=42)
perm.fit(X_test_scaled, y_test)
eli5.show_weights(perm, feature_names=X_test_df.columns.tolist())
```

Résultat obtenu :

Weight	Feature
0.1221 ± 0.0362	Glucose
0.0312 ± 0.0301	BMI
0.0065 ± 0.0184	BloodPressure
0.0039 ± 0.0226	DiabetesPedigreeFunction
-0.0039 ± 0.0324	Pregnancies
-0.0156 ± 0.0226	SkinThickness
-0.0169 ± 0.0194	Age
-0.0286 ± 0.0156	Insulin

FIGURE 3.20 – Importance des variables selon ELI5

Analyse synthétique : La variable `Glucose` ressort comme la plus influente dans la prédiction du diabète. D'autres variables comme `BMI` ont un effet plus modéré, tandis que certaines variables présentent un impact faible ou négatif. Ces résultats confirment les tendances observées via les autres méthodes d'explicabilité.

Explication par décomposition de Taylor sur le dataset Breast Cancer Wisconsin

Nous avons appliqué la méthode *Deep Taylor Decomposition* pour interpréter les prédictions d'un réseau de neurones entraîné sur le jeu de données `Breast Cancer`. Cette méthode permet de quantifier l'importance des variables en décomposant la sortie du modèle sous forme de contributions attribuées à chaque feature.

Préparation des données Pour chaque instance, nous avons normalisé les variables et extrait une prédiction du modèle à expliquer. L'instance testée ici correspond à un échantillon de la classe maligne.

Listing 3.6 – Application de Deep Taylor

Code essentiel

```
analyzer = innvestigate.create_analyzer("deep_taylor", model)
instance = X_test_scaled[0:1]
relevance = analyzer.analyze(instance)
plt.barh(data.feature_names, relevance[0])
plt.title("Explication locale  Deep Taylor")
plt.xlabel("Pertinence")
```

Analyse de l'explication locale Le graphique ci-dessus montre les valeurs de pertinence attribuées à chaque feature pour une instance donnée. Les variables ayant eu le plus d'influence sur la prédiction du modèle sont :

- **mean radius** : la variable la plus influente, indiquant que la taille moyenne de la tumeur est fortement corrélée à la malignité.
- **mean area** et **worst smoothness** : également très contributives.
- **mean concave points** et **fractal dimension error** : impact modéré.

Les autres caractéristiques (telles que *radius error*, *worst compactness*) ont un poids négligeable dans cette instance particulière.

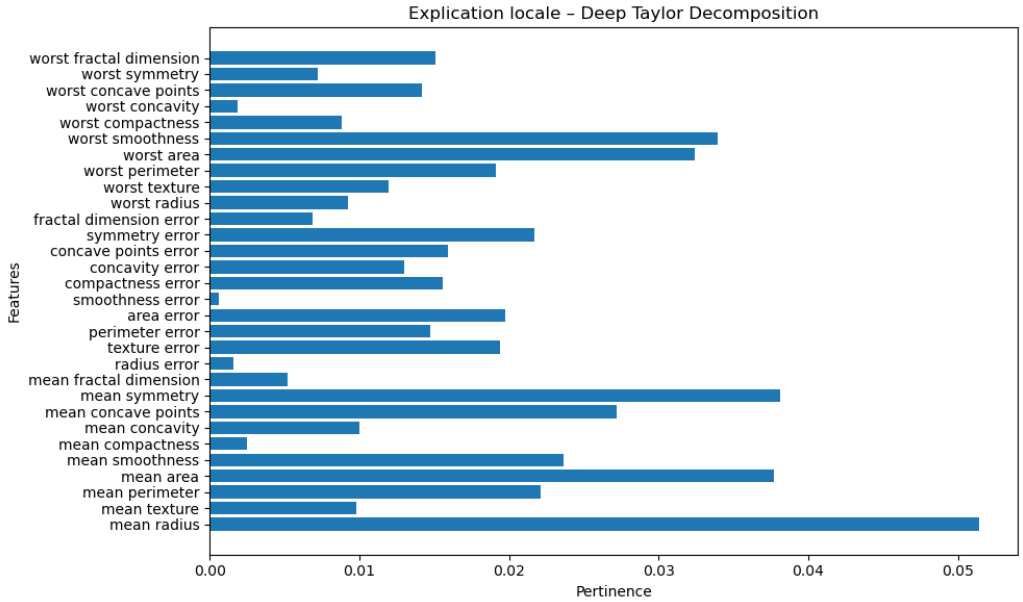

FIGURE 3.21 – Explication locale avec Deep Taylor pour une instance du dataset Breast Cancer

Pertinence globale Pour obtenir une vision globale, nous avons moyenné les valeurs de pertinence sur les dix premières instances du jeu de test.

Listing 3.7 – Pertinence moyenne globale

```
global_relevance = np.mean([analyzer.analyze(x.reshape(1, -1))
                            for x in X_test_scaled[:10]], axis=0)
plt.barh(data.feature_names, global_relevance.flatten())
plt.title("Pertinence moyenne (globale)")
```

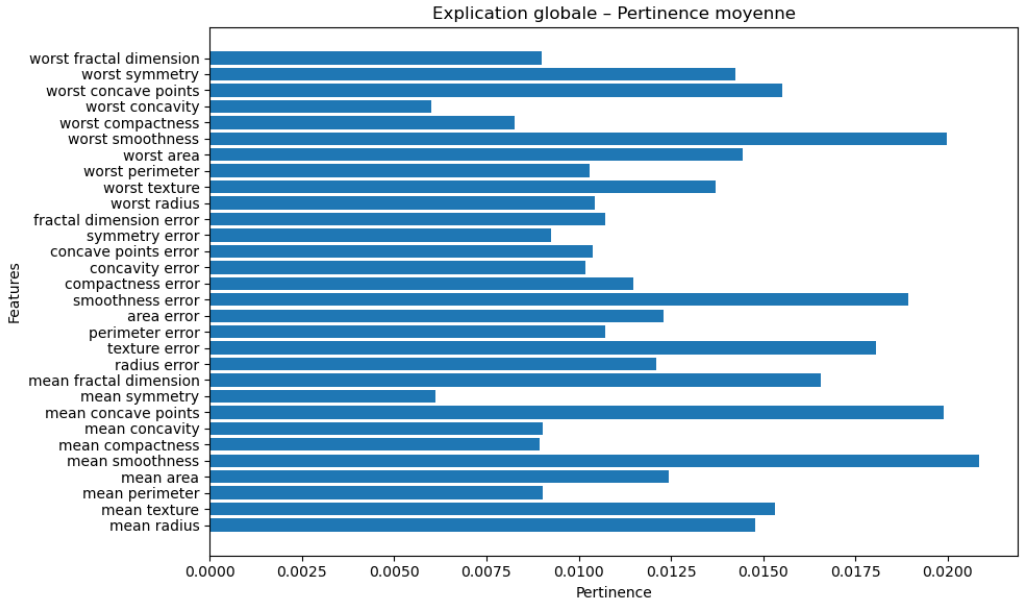

FIGURE 3.22 – Explication globale – Moyenne des pertinences sur 10 instances

Analyse de l'explication globale L'analyse globale confirme l'importance de certaines caractéristiques clés :

- **mean concave points**, **worst smoothness**, **mean area** : ce sont les variables les plus pertinentes pour la majorité des prédictions.
- **mean symmetry** et **worst compactness** : impact intermédiaire.
- D'autres variables comme *texture error* ou *perimeter error* apparaissent moins influentes.

Conclusion La méthode de décomposition de Taylor met en évidence des patterns pertinents sur le plan clinique : les mesures liées à la forme, la taille et la régularité de la tumeur dominent dans les décisions du modèle. Ce résultat renforce la confiance dans l'interprétabilité de notre réseau de neurones appliqué à ce dataset.

LIME

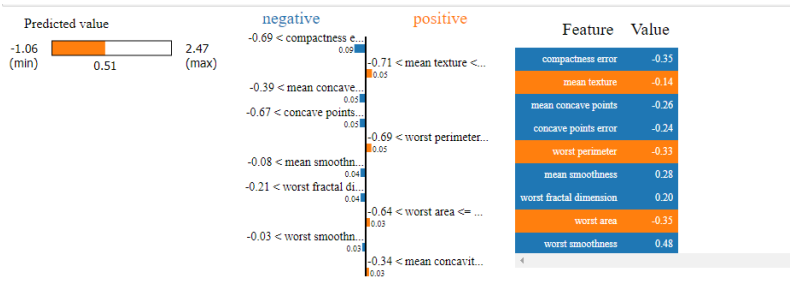

FIGURE 3.23 – Explication locale par LIME pour une instance du jeu de données Breast Cancer

L'explication générée par LIME met en évidence les attributs qui influencent localement la prédiction du réseau de neurones pour une instance spécifique. Dans cet exemple, la prédiction est proche de la frontière de décision (valeur prédite = 0,51), traduisant une certaine incertitude du modèle.

Les caractéristiques ayant contribué à éloigner la prédiction de la classe maligne sont principalement :

- Une faible compacité (*compactness error* < -0,69),
- Une texture moyenne réduite (*mean texture* < -0,71),
- Une faible présence de concavité (*mean concave points*, *concave points error*),
- Un périmètre faible (*worst perimeter*),
- Une faible surface maximale (*worst area*).

À l'inverse, quelques variables ont eu un effet modérément positif sur la prédiction, comme :

- La *worst smoothness* relativement élevée,
- La *mean smoothness* au-dessus de la moyenne.

Cette interprétation locale permet de mieux comprendre la manière dont le modèle pondère les différents attributs morphologiques de la tumeur, en particulier la régularité des contours et la forme globale, pour émettre une prédiction individualisée.

Force Plot

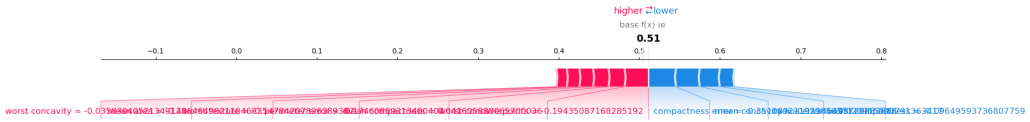

FIGURE 3.24 – Visualisation SHAP : Force Plot pour une instance du jeu de données Breast Cancer

Le *Force Plot* est une représentation visuelle générée par SHAP qui met en évidence la manière dont chaque variable contribue à la prédiction d'un modèle pour une instance particulière. L'axe horizontal représente la valeur de sortie du modèle (probabilité de malignité), tandis que chaque barre colorée correspond à l'effet d'une variable.

- La **base value** (valeur de référence) est ici de 0,51. Elle correspond à la sortie moyenne du modèle sans tenir compte des variables spécifiques à l'instance.
- Les **barres bleues** indiquent les variables qui **augmentent** la probabilité de malignité.
- Les **barres rouges** désignent les variables qui **diminuent** cette probabilité.
- La **prédiction finale** (ligne verte) est le résultat de l'ajustement progressif à partir de la base value, après prise en compte de toutes les contributions.

Dans cette instance, la prédiction finale est de 0,51, ce qui traduit un équilibre entre facteurs aggravants et rassurants. Deux variables se démarquent :

- **Compactness** : cette caractéristique possède une valeur relativement élevée (0,194) pour cette patiente et contribue fortement à **augmenter** la probabilité de malignité (longue barre bleue).
- **Worst concavity** : avec une valeur faible (-0,036), elle contribue légèrement à **réduire** la prédiction (courte barre rouge).

Ce graphique met en lumière le rôle crucial de certaines variables dans la décision du modèle. Il constitue un outil précieux pour l'interprétation locale, en permettant de justifier les prédictions auprès de professionnels de santé de manière transparente.

LIME sur le jeu de données Cleveland Heart Disease

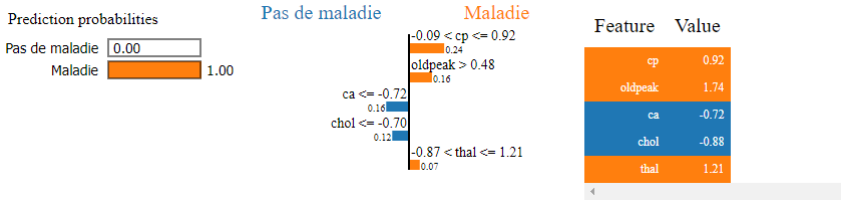

FIGURE 3.25 – Visualisation LIME : contribution locale des attributs à la prédiction d'un patient dans le dataset Cleveland Heart Disease.

L'explication fournie par LIME dans la figure 3.25 permet d'identifier les variables ayant le plus influencé la prédiction d'un cas de maladie cardiaque. Les attributs les plus déterminants sont :

— **cp (type de douleur thoracique)** : une valeur élevée (0.92, normalisée) a fortement contribué à prédire une maladie.
— **oldpeak (dépression ST)** : une valeur significative (1.74) a également renforcé cette prédiction.
— **ca (nombre de vaisseaux colorés)** : bien que sa valeur soit faible (≤ -0.72), elle a eu un poids positif dans la détection de la maladie.
— **chol (cholestérol)** : une valeur basse (≤ -0.88) a légèrement soutenu le diagnostic.
— **thal (valeur de thalassemie)** : bien que sa contribution soit plus modeste, elle reste significative.

Ainsi, LIME met en évidence que ce cas est principalement influencé par les signes cliniques visibles, tels que l'intensité de la douleur thoracique et l'élévation du segment ST à l'effort. Ces facteurs ont orienté le modèle vers une prédiction positive de maladie cardiaque.

Waterfall Plot SHAP sur le jeu de données Cleveland Heart Disease

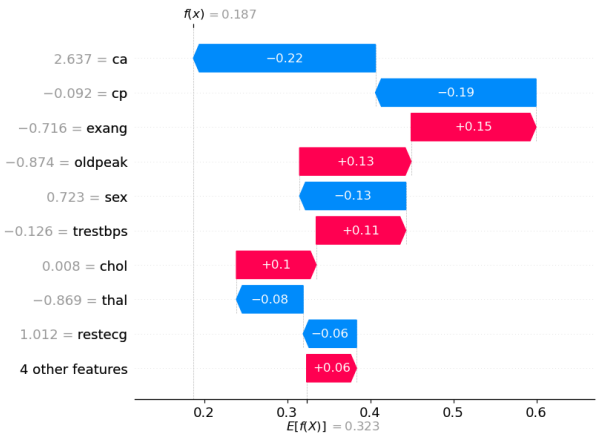

FIGURE 3.26 – Waterfall plot SHAP illustrant la contribution des features à la prédiction pour une instance du jeu de données Cleveland Heart Disease.

Le *Waterfall Plot* présenté en figure 3.26 permet de visualiser comment chaque variable influence la prédiction finale du modèle pour une observation donnée. La **valeur de base** est de 0,323, correspondant à la probabilité moyenne de maladie cardiaque sur l'ensemble des données. La prédiction finale pour cette instance est de 0,187, indiquant une probabilité relativement faible de pathologie.

Les contributions les plus marquantes sont les suivantes :

- **ca** (nombre de vaisseaux coronaires visibles à l'angiographie) : valeur élevée (2,637), ce qui diminue fortement la probabilité de maladie. C'est la variable la plus protectrice ici.
- **exang** (angine induite par l'effort) : valeur négative, effet positif fort sur la probabilité, donc facteur de risque important.
- **cp** (type de douleur thoracique), **sex** (sexe), **thal**, **restecg** : impactent négativement la probabilité, bien que de manière modérée.
- **oldpeak**, **trestbps**, **chol** : contribuent positivement mais faiblement à la prédiction de maladie.

Les barres rouges indiquent les variables qui augmentent la probabilité de maladie cardiaque, tandis que les barres bleues représentent celles qui la diminuent. La longueur de chaque barre reflète l'importance relative de la contribution.

En résumé, ce graphique met en évidence un équilibre entre des facteurs de protection (notamment le nombre de vaisseaux coronaires visibles) et des facteurs de risque (principalement l'effort et la dépression ST), justifiant la probabilité modérée estimée par le modèle pour cette instance.

3.4 Comparaison des approches explicatives

L'analyse croisée des méthodes d'explicabilité appliquées aux données tabulaires (LIME, SHAP) et aux données visuelles (Grad-CAM, etc.) permet de dégager plusieurs enseignements. Ces approches ont été comparées selon trois critères principaux : la lisibilité des résultats pour un utilisateur non spécialiste, la pertinence des explications vis-à-vis du contexte médical, et enfin la complexité technique de mise en œuvre.

Lisibilité : Les méthodes comme LIME ou le Summary Plot SHAP offrent des représentations intuitives pour les données tabulaires, permettant de relier facilement chaque variable à son influence sur la prédiction. En revanche, certaines visualisations (force plot, waterfall plot) exigent un certain niveau de familiarité avec la logique des SHAP values pour être pleinement comprises. Concernant les images médicales, Grad-CAM ou LRP fournissent des cartes de chaleur directement interprétables par un clinicien, ce qui constitue un avantage considérable en termes de communication.

Pertinence des explications : SHAP s'est montré particulièrement robuste pour mettre en évidence des variables reconnues comme significatives par les professionnels de santé (par exemple, le nombre de vaisseaux coronaires ou le type de douleur thoracique). Les méthodes visuelles ont quant à elles révélé des régions pertinentes dans les images médicales, mais leur interprétation dépend fortement de la qualité des données et du modèle sous-jacent.

Complexité et temps de calcul : Les approches tabulaires comme LIME sont relativement simples à déployer, même si elles nécessitent un échantillonnage local important. SHAP, notamment avec `KernelExplainer`, peut s'avérer coûteux en temps de calcul, surtout pour des modèles non linéaires. Du côté des images, les méthodes comme Grad-

CAM exigent une manipulation fine du graphe computationnel du réseau de neurones, ce qui peut freiner leur intégration dans des environnements hospitaliers à forte contrainte opérationnelle.

En résumé, si les méthodes explicatives s'avèrent utiles dans la compréhension des prédictions, leur intégration réelle dans un cadre clinique nécessite de prendre en compte à la fois la clarté des résultats, leur utilité pour la prise de décision médicale, et la faisabilité technique de leur déploiement.

3.5 Limites et perspectives

L'étude menée met en évidence plusieurs limites dans l'application des méthodes d'explicabilité, en particulier lorsque celles-ci sont confrontées à des données réelles issues du domaine médical.

Limites identifiées : Certaines visualisations, notamment les plots SHAP pour les jeux de données à faible effectif, manquent de densité ou de lisibilité. Le nombre limité d'instances empêche parfois d'identifier des tendances générales ou de dégager des profils de risque fiables. De plus, l'interprétation reste partielle pour les modèles à architecture complexe, où les effets d'interaction entre variables peuvent brouiller la compréhension des résultats fournis par les méthodes d'explication.

Pistes d'amélioration :

— *Enrichissement des données* : Augmenter le nombre de cas cliniques analysés, tant en termes de patients que d'images médicales, afin d'améliorer la robustesse des explications.

— *Approches multimodales* : Intégrer conjointement les données visuelles et tabulaires dans un cadre d'apprentissage multimodal, plus fidèle à la complexité des décisions médicales.

— *Exploration de nouvelles méthodes* : Expérimenter des techniques explicatives avancées telles que LRP, RISE ou XRAI, susceptibles d'apporter des explications plus localisées ou plus stables.

— *Validation clinique* : Impliquer systématiquement des experts médicaux dans l'évaluation des explications pour s'assurer de leur pertinence et favoriser leur adoption dans la pratique.

Ces axes d'amélioration peuvent contribuer à renforcer l'impact des méthodes d'explicabilité dans le domaine médical, en les rapprochant des besoins réels du terrain clinique.

Synthèse du chapitre

Ce chapitre a permis d'examiner concrètement le comportement des modèles de réseaux de neurones à travers diverses méthodes d'explicabilité, appliquées à plusieurs jeux de données médicaux. Les visualisations générées ont mis en évidence la complémentarité des approches : les méthodes comme SHAP et LIME ont fourni des interprétations détaillées sur les données tabulaires, tandis que les techniques visuelles telles que Grad-CAM ou Taylor ont permis d'expliquer les prédictions sur des images médicales.

Il ressort de ces analyses que la combinaison de plusieurs outils explicatifs est essentielle pour obtenir une compréhension plus complète et fiable des décisions du modèle. Chaque méthode apporte une perspective différente, utile selon le type de données et le

niveau d'analyse souhaité (globale ou locale). Ce croisement d'explications se révèle particulièrement pertinent dans le domaine médical, où la transparence et la justification des décisions sont indispensables.

Les résultats obtenus confirment ainsi l'importance de l'explicabilité dans le contexte de l'IA en santé, non seulement pour évaluer la pertinence des modèles, mais aussi pour instaurer un climat de confiance autour de leur usage futur en environnement clinique.

Conclusion

> *« You can't just look at numbers and say, "Trust me." We need explanations we can understand. »*
>
> — Judea Pearl, pionnier de l'intelligence causale

À l'issue de ce travail, il apparaît avec évidence que la puissance des réseaux de neurones, aussi impressionnante soit-elle, ne peut suffire à garantir une adoption sereine et responsable de l'intelligence artificielle dans les domaines sensibles comme la médecine. Si ces modèles sont aujourd'hui capables d'atteindre des performances comparables, voire supérieures à celles des experts humains, leur complexité intrinsèque pose une question fondamentale : comment faire confiance à une décision que l'on ne peut pas expliquer ?

C'est dans cette tension entre efficacité prédictive et besoin de transparence que s'inscrit l'explicabilité, au cœur de notre étude. Nous avons, dans un premier temps, posé les bases théoriques des réseaux de neurones, en détaillant leur fonctionnement, leur architecture, et les défis associés à leur entraînement et leur évaluation. Ce socle nous a permis de mieux comprendre pourquoi ces modèles sont à la fois si performants... et si opaques.

Dans un second temps, nous avons exploré les différentes méthodes développées pour lever, au moins partiellement, le voile sur ces systèmes complexes. Qu'il s'agisse de techniques locales comme LIME ou SHAP, de visualisations basées sur les gradients comme Grad-CAM ou Integrated Gradients, ou encore de méthodes globales comme les Partial Dependence Plots ou les modèles substituts, chaque approche offre un éclairage complémentaire sur la façon dont le modèle « pense » et décide.

Enfin, l'application de ces méthodes à des jeux de données médicaux, tant visuels que tabulaires, a permis de confronter la théorie à la pratique. Nous avons pu observer concrètement les apports — mais aussi les limites — de ces techniques d'explication : certaines offrent une lecture intuitive, d'autres une granularité plus fine, mais toutes nécessitent un regard critique pour être correctement interprétées. Cette étape a été essentielle pour évaluer la pertinence de chaque méthode dans un contexte clinique réel, où l'objectif n'est pas seulement de comprendre la machine, mais d'aider le praticien à décider en connaissance de cause.

Ce mémoire ne prétend pas apporter de solution universelle, mais il invite à une réflexion essentielle sur la place que nous souhaitons accorder à l'intelligence artificielle dans la décision médicale. Une IA n'est pas une vérité absolue, mais un outil d'aide — et cet outil doit rester lisible, contestable, améliorable.

Les perspectives de ce travail sont nombreuses. D'un point de vue technique, l'exploration de méthodes explicatives plus avancées ou la combinaison multimodale des données (images et tabulaires) ouvrent de nouvelles pistes. D'un point de vue humain, l'implication directe des professionnels de santé dans l'analyse et l'évaluation des explications pourrait renforcer la validité clinique des modèles. Enfin, d'un point de vue éthique, ce travail

rappelle l'importance de concevoir des systèmes d'IA qui ne soient pas seulement performants, mais aussi responsables, transparents et alignés avec les besoins des utilisateurs finaux.

En somme, l'explicabilité n'est pas une simple option technique, mais un impératif scientifique et sociétal. Elle constitue une passerelle entre les algorithmes et l'humain, entre la prédiction et la compréhension, entre l'innovation et la confiance.

Bibliographie

[1] Walaa Al-Dhabyani, Mohamed Gomaa, Heba Khaled, and Amr Fahmy. A dataset of breast ultrasound images for the study of breast cancer. *Data in Brief*, 28 :104863, 2020.

[2] Suresh Beekhani. Top 10 activation functions in deep learning. https://www.linkedin.com/pulse/top-10-activation-functions-deep-learning-suresh-beekhani-vbisf, 2024. Consulté en mai 2025.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] Codewave. History and development of neural networks in ai. https://codewave.com/insights/development-of-neural-networks-history/, 2024. Consulté en mai 2025.

[5] Redress Compliance. The history of neural networks in ai. https://redresscompliance.com/the-history-of-neural-networks-in-ai/, 2025. Consulté en mai 2025.

[6] Wikipédia contributors. Perceptron multicouche - wikipédia. https://fr.wikipedia.org/wiki/Perceptron_multicouche, 2024. Consulté en mai 2025.

[7] Data Franca. Perceptron multicouche - datafranca. https://datafranca.org/wiki/Perceptron_multicouche, 2025. Consulté en mai 2025.

[8] DataCamp. Introduction to activation functions in neural networks. https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks, 2024. Consulté en mai 2025.

[9] DataScientest. Convolutional neural network (cnn) : Définition, fonctionnement, utilisation, 2022. URL https://datascientest.com/convolutional-neural-network. Consulté en juin 2025.

[10] R. et al. Detrano. Heart disease data set (cleveland). https://archive.ics.uci.edu/ml/datasets/Heart+Disease, 1988. UCI Machine Learning Repository.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016.

[12] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, 2019.

[13] IBM. Architecture (perceptron multicouche). https://www.ibm.com/docs/fr/spss-statistics/saas?topic=perceptron-architecture-multilayer, 2024. Consulté en mai 2025.

[14] V7 Labs. Activation functions in neural networks [12 types & use cases]. https://www.v7labs.com/blog/neural-networks-activation-functions, 2023. Consulté en mai 2025.

[15] Warren McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5 :115–133, 1943.

[16] Marvin Minsky and Seymour Papert. *Perceptrons : An Introduction to Computational Geometry*. MIT Press, 1969.

[17] Christoph Molnar. *Interpretable Machine Learning : A Guide for Making Black Box Models Explainable*. Self-published, 2 edition, 2022. Available online at https://christophm.github.io/interpretable-ml-book/.

[18] Andrew Ng. Deep learning specialization. https://www.coursera.org/specializations/deep-learning, 2024. Consulté en mai 2025.

[19] Erik Roberts. Neural networks - history. https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/history1.html, n.d. Consulté en mai 2025.

[20] Frank Rosenblatt. The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65 :386–408, 1958.

[21] Fabrice Rossi. Réseaux de neurones : le perceptron multi-couches. https://apiacoa.org/publications/teaching/nn/MLP.pdf, 2023. Consulté en mai 2025.

[22] Towards Data Science. Activation functions in neural networks : How to choose the right one. https://towardsdatascience.com/activation-functions-in-neural-networks-how-to-choose-the-right-one-cb20414c04e5, 2023. Consulté en mai 2025.

[23] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam : Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.

[24] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks : Visualising image classification models and saliency maps. *arXiv preprint arXiv :1312.6034*, 2013.

[25] Leonhard Sixt and Tim Landgraf. A rigorous study of the deep taylor decomposition. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=IUhEqnntDx.

[26] Jack W Smith, John E Everhart, William C Dickson, William C Knowler, and R Scott Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. In *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265, 1988.

[27] W.N. Street, W.H. Wolberg, and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *Biomedical Image Processing and Biomedical Visualization*, volume 1905, pages 861–870. SPIE, 1993.

[28] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3319–3328, 2017.

[29] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset : A large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5 :180161, 2018.

[30] Dave UIPUC. Types of activation functions : Sigmoid, tanh, relu, softmax. https://www.linkedin.com/pulse/types-activation-functions-sigmoid-tanh-relu-softmax-part-dave-uipuc, 2024. Consulté en mai 2025.

[31] Wikipedia contributors. Perzeptron - wikipedia. https://de.wikipedia.org/wiki/Perzeptron, 2024. Consulté en mai 2025.

[32] Quantum Zeitgeist. Mcculloch–pitts neuron : A look at the foundation of the artificial neuron. https://quantumzeitgeist.com/mcculloch-pitts-neuron-a-look-at-the-foundation-of-the-artificial-neuron/, 2025. Consulté en mai 2025.

Annexe A

Compléments d'analyse visuelle

A.1 Analyse des images explicatives – DermaMNIST

L'image analysée présente une lésion cutanée en niveaux de gris, avec trois cartes d'explicabilité générées par un modèle CNN : Integrated Gradients (IG), Saliency Maps et Grad-CAM. Ces cartes révèlent les zones ayant le plus influencé la décision du modèle.

- **Image originale :** La lésion a une forme irrégulière avec un centre foncé. Sans les surbrillances, il est difficile de savoir quelles zones sont pertinentes.
- **Integrated Gradients :** Met en évidence le centre de la lésion comme zone déterminante, suggérant que la texture et l'intensité de cette région guident la décision.
- **Saliency Maps :** Confirme l'importance du centre, avec des gradients élevés indiquant une forte sensibilité de la prédiction à cette zone.
- **Grad-CAM :** Localise l'activation principale au centre également, montrant la cohérence entre les couches profondes du réseau et les autres méthodes.

Conclusion : Le modèle semble capter des caractéristiques cliniquement pertinentes comme l'asymétrie, les contours irréguliers ou l'hétérogénéité de texture.

A.2 Analyse des images explicatives – BreastMNIST

Cette section reprend la même analyse appliquée à une mammographie issue du dataset BreastMNIST.

- **Image originale :** La mammographie montre une masse sombre suspecte.
- **Integrated Gradients :** Active intensément le centre de la masse, suggérant que sa densité et sa texture sont cruciales pour la prédiction.
- **Saliency Maps :** Accentue les variations au centre, avec des contours faiblement soulignés.
- **Grad-CAM :** Identifie la même région centrale comme la plus activée, montrant la cohérence de la représentation du réseau.

Conclusion : Le modèle semble apprendre des motifs compatibles avec les critères médicaux (densité, texture, contours), validant en partie sa fiabilité dans un contexte clinique