République Algérienne Démocratique et Populaire
Université Abderrahmane MIRA de Béjaïa
Faculté des Sciences Exactes

Département de Recherche Opérationnelle

Mémoire présenté pour l'obtention du diplôme de Master
en Mathématiques Appliquées

Spécialité : Sciences de Données et Aide à la Décision

Méthodes Random Forest et LSTM pour la prévision des matières premières au sein de l'entreprise BATELEC.

Présenté par :
AMROUCHE Fouad
HASSANI Abderrahim Amine

Défendu le 29/06/2025 devant le jury composé de :

Mlle Z.AOUDIA	M.C. classe A	Présidente du jury	UAMB - Béjaïa
Mr R.SAHLI	Doctorant	Examinateur	UAMB - Béjaïa
Mr B.BRAHMI	M.C. classe B	Examinateur	UAMB - Béjaïa
Mr L.BOUZIDI	M.C. classe B	Examinateur	UAMB - Béjaïa
Mr S.TOUATI	M.C. classe B	Encadreur	UAMB - Béjaïa

Année Universitaire 2024–2025

Dédicace

À nos parents ,nos familles et nos amis
pour leur patience et leur encouragement tout au long de ce parcours.

À notre encadrant,
pour sa précieuse aide et ses encouragements dans les moments de doute.

À tous ceux qui ont contribué, de près ou de loin,
à la réalisation de ce travail.

Remerciements

Nos remerciements vont en particulier à notre encadrant, **Mr S. TOUATI**, pour sa disponibilité, ses conseils éclairés, son accompagnement rigoureux et son engagement constant tout au long de ce travail.

Nous exprimons également notre profonde reconnaissance à **Mr Nadir Bekka**, responsable de la planification chez BATELEC, pour son aide précieuse, sa collaboration et sa disponibilité tout au long de notre étude de cas.

Nous remercions les membres du jury, **Mlle Z. AOUDIA**, **Mr B. BRAHMI**, **Mr L. BOUZIDI** et **Mr R. SAHLI**, pour l'honneur qu'ils nous font en acceptant d'évaluer ce mémoire.

Nos pensées les plus respectueuses vont à l'ensemble du corps enseignant du département de Recherche Opérationnelle de l'Université de Béjaïa, pour la qualité de leur enseignement et leur contribution à notre formation académique.

Enfin, nous adressons nos remerciements les plus chaleureux à nos familles et amis pour leur soutien indéfectible, leur patience et leurs encouragements, qui nous ont accompagnés tout au long de ce parcours universitaire.

Table des matières

Table des figures

Liste des tableaux

Liste des symboles

x	Donnée d'entrée
y	Variable cible (valeur observée)
$\hat{y}$	Valeur prédite
X	Matrice des variables explicatives
Y	Vecteur des cibles
$f(x)$	Fonction de prédiction du modèle
β_j	Coefficient de régression pour la variable x_j
RSS	Somme des carrés des résidus (Résidu quadratique total)
$L(f, X, Y)$	Fonction de perte du modèle f sur les données (X, Y)
ℓ_t	Niveau estimé à l'instant t (lissage exponentiel)
b_t	Tendance estimée à l'instant t
s_t	Composante saisonnière à l'instant t
α, β, γ	Coefficients de lissage
Φ_i	Valeur de Shapley pour la caractéristique i
T_t	Tendance dans une série temporelle
S_t	Saison dans une série temporelle

ε_t	Résidu ou bruit aléatoire
C_k	Cluster numéro k
μ_k	Centroïde du cluster k
$\|\|x_i - \mu_k\|\|^2$	Distance euclidienne entre x_i et μ_k
y_t	Valeur observée d'une série temporelle à l'instant t
$\hat{y}_{t+h}$	Prévision à h périodes dans le futur

Liste des abréviations

AI	Intelligence Artificielle (Artificial Intelligence)
CNN	Réseau de neurones convolutifs (Convolutional Neural Network)
FI	Importance des caractéristiques (Feature Importance)
GSA	Analyse de sensibilité globale (Global Sensitivity Analysis)
LIME	Local Interpretable Model-agnostic Explanations
LSTM	Long Short-Term Memory
ML	Machine Learning (Apprentissage automatique)
MR	Model Reliance
RNN	Réseau de neurones récurrents (Recurrent Neural Network)
SHAP	SHapley Additive exPlanations
XAI	Intelligence Artificielle Explicable (eXplainable AI)

Introduction générale

L'ère numérique a été marquée par une prolifération sans précédent de données, transformant profondément les domaines de la science, de l'ingénierie et des affaires. Au cœur de cette transformation se trouve le domaine de l'apprentissage automatique (Machine Learning), qui, grâce à des algorithmes sophistiqués, permet aux systèmes d'apprendre à partir de ces données et de prendre des décisions ou de faire des prédictions. Des applications allant de la reconnaissance d'images à la prévision météorologique, en passant par le diagnostic médical et la finance, témoignent de l'impact révolutionnaire du machine learning sur notre quotidien.

Cependant, à mesure que les modèles de machine learning deviennent de plus en plus complexes, notamment avec l'avènement des réseaux de neurones profonds, leur fonctionnement interne tend à devenir opaque. Cette "boîte noire" pose des défis significatifs, en particulier dans les domaines critiques où la transparence, la confiance et la responsabilité sont primordiales. Comprendre pourquoi un modèle prend une certaine décision, plutôt que de simplement savoir quelle décision il prend, est devenu une exigence fondamentale. Sans cette compréhension, il est difficile de valider la robustesse des modèles, d'identifier et de corriger les biais potentiels, d'assurer la conformité réglementaire, et de bâtir la confiance des utilisateurs.

Ce mémoire s'inscrit dans ce contexte, avec pour objectif principal d'étudier les approches permettant de rendre les modèles de machine learning plus compréhensibles, sans compromettre leur performance. Pour cela, nous adoptons une démarche en deux volets : d'une part, une présentation des fondements théoriques du machine learning, de ses modèles phares (tels que la régression linéaire, Random Forest, LSTM) et de la problématique d'interprétabilité ; d'autre part, une application concrète sur des données industrielles issues de l'entreprise BATELEC, spécialisée dans les équipements électriques.

À travers cette étude de cas, nous illustrons l'utilisation de modèles prédictifs appliqués à la prévision de la consommation de matières premières, et leur interprétation à l'aide d'outils tels que SHAP et l'importance des variables explicatives . Ces outils permettent non seulement d'identifier les variables influentes, mais aussi de justifier les décisions prises par les modèles, contribuant ainsi à une meilleure compréhension, transparence et confiance dans les systèmes automatisés.

CHAPITRE 1

Les Fondements de la science de données et de l'apprentissage automatique

Introduction

L'évolution récente de l'apprentissage informatique et de la science des données a changé en profondeur notre manière de collecter, d'analyser et de traiter les informations. Avec le volume de données généré chaque jour, la capacité à en extraire une valeur décisionnelle pertinente est devenue quasi indispensable pour un grand nombre de secteurs, allant de l'industrie à la finance, en passant par la santé et la logistique.

L'objectif avancé dans ce chapitre consiste à expliquer pas à pas les principes élémentaires du machine learning et de la science des données. Au début, nous expliquons ce qu'on appelle donnée et information tout en insistant sur leur typologie et leur qualité. Il expose ensuite les principales familles d'apprentissage (supervisé, non supervisé, par renforcement, par transfert), ainsi que les modèles emblématiques associés, tels que la régression linéaire, les forêts aléatoires (Random Forest), ou encore les réseaux de neurones à mémoire longue (LSTM).

Enfin, une attention particulière est portée sur les séries temporelles, un domaine spécifique d'analyse où les notions de temporalité, de tendance et de saisonnalité sont cruciales. Des méthodes traditionnelles comme les modèles

à base de lissage y sont également abordées, en tant que préambule aux techniques modernes de prévision.

1.1 Science de données

La science des données est un domaine multidisciplinaire qui englobe les métiers tels que l'informatique, les mathématiques, la statistique et le secteur d'activité en question dans le but d'extraire des informations pertinentes à partir de données volumineuses, [29], qu'elles soient en format structuré ou non.Cela comprend chaque étape du cycle de données : la collecte, le nettoyage, l'analyse exploratoire, le modelage prédictif par le biais de machine learning, la visualisation et la prise de décision.

Aujourd'hui, la liste d'industries qui supplémentent leurs opérations par la science des données va au-delà de la finance pour inclure la santé, le transport et même le marketing. Cela a permis l'automatisation de certaines tâches, une meilleure anticipation des comportements, et même l'optimisation de systèmes complexes.

1.2 L'information et la donnée

1.2.1 Définition d'une donnée

la définition d'une donnée fait généralement l'unanimité. D'après T.H Davenport et L.Prusak, une donnée est un élément brut, sans signification propre, qui décrit des faits, des observations ou des mesures. Elle acquiert du sens lorsqu'elle est traitée, structurée ou interprétée dans un contexte donné, [8].
Et une donnée est un fait, un concept ou un objet réel, voire imaginaire, qui a été codé de manière à être traité, stocké ou communiqué par un système informatique. C'est une matière première de l'information, souvent vide de signification jusqu'à ce qu'on l'interprète, d'après R.Y Wang et D.M Strong, [40].

À partir des deux définitions présentées, on peut conclure que la donnée constitue l'élément fondamental, une matière première en soi, dépourvue de si-

gnification lorsqu'elle est isolée. Elle représente des faits bruts, des observations ou des mesures, codés de manière à pouvoir être stockés, traités et communiqués par des systèmes informatiques. Ce n'est qu'une fois mise en contexte, interprétée ou structurée qu'elle devient réellement utile et contribue à la production d'une information.

On peut classer les données selon plusieurs critères parmi eux :

1.2.1.1 Selon le type

1. **Données structurées :** ce sont des Données organisées selon un format fixe, souvent stockées dans des bases relationnelles, [19], par exemple Une base de données clients (nom, prénom, e-mail, téléphone) ou un tableau Excel .

2. **Données semi-structurées :**ce sont des Données qui ne suivent pas une structure rigide comme les bases relationnelles, mais qui possèdent tout de même une organisation hiérarchique, [1] comme des fichiers XML ou JSON.

3. **Données non structurées :** ce sont des données exprimées en langage naturel et sans structure spécifique ni type de domaine n'est défini, [35], pour exemple : des documents texte (PDF, Word), images, vidéos ou enregistrements audio.

4. **Données qualitatives et quantitatives :** les donnes qualitatives sont des données descriptives et non numérique (opinions d'utilisateurs, commentaires) contrairement au donnes quantitatives qui sont des donnes numérique e et mesurable (Salaires, chiffre d'affaires)

1.2.1.2 Selon la qualité

D'après R.Y Wang et D.M Strong, les données peuvent être évaluées et classées selon plusieurs dimensions de qualité.

1. **Précision :** C'est le Degré auquel une donnée reflète correctement la réalité ou la valeur réelle.
2. **Complétude :** Mesure dans laquelle toutes les valeurs nécessaires pour une analyse ou une recherche sont présentes(Un formulaire client avec tous les champs remplis).
3. **Actualité :** La disponibilité des Données au bon moment pour la prise de décision.
4. **Cohérence :** c'est l'absence de contradiction entre différentes sources ou systèmes(Un même client ne doit pas avoir des adresses différentes).
5. **Traçabilité / Source :** Capacité à retracer l'origine des données et leur parcours.

1.3 Machine Learning

il existe plusieurs définitions pour le machine learning ou l'apprentissage automatique en français parmi ces définitions, on retrouve :

Définition 1. Machine Learning est la science (et l'art) de programmer des ordinateurs afin qu'ils puissent apprendre à partir des données.

-Géron[12].

Définition 2. Le machine learning s'intéresse à la manière de concevoir des programmes informatiques qui s'améliorent automatiquement grâce à l'expérience.

–Russell et Norvig[31].

Définition 3. Un programme informatique apprend à partir d'une expérience E, pour une classe de tâches T et une mesure de performance P, si sa performance dans les tâches T, mesurée par P, s'améliore avec l'expérience E.

- Tom M and Mitchell [24]

Toutes ces définitions soulignent que le Machine Learning est une méthode d'apprentissage automatique, où un système informatique utilise des données et apprend à reconnaître des modèles ou des règles, pour améliorer ses performances ou faire des prédictions sans être explicitement programmé pour chaque tâche.

Si on revient sur la définition 3 d'un point de vue mathématique, le fait de s'améliorer c'est de réduire l'erreur , il y a toujours une question d'erreur en machine learning, ça peut être l'écart entre un point (une donnée) et une droite de régression , ou la différence entre une valeur prédite et la vraie valeur, c'est toujours une question d'erreur et le but est toujours de la minimiser.

On peut ainsi modéliser le problème tel que :

- Un ensemble de données d'entrée : $X = \{x_1, x_2, \ldots, x_n\}$, où $x_i \in \mathbb{R}^p$;
- Un ensemble de données de sortie : $Y = \{y_1, y_2, \ldots, y_n\}$, où $y_i \in \mathbb{R}$;
- Une fonction $f : X \to Y$, appelée *modèle*, qui s'ajuste aux données en minimisant l'erreur du modèle $L(f, X, Y)$, tel que
  $L(f, X, Y)$ une fonction de perte(ou fonction de coût).

1.4 Typologie du Machine Learning

La typologie du Machine Learning se divise en plusieurs grandes catégories selon la nature des données et la manière dont l'apprentissage est réalisé, chacune avec ses avantages et ses inconvénients. Parmi ces typologies, on retrouve :

1.4.1 Apprentissage supervisé

En apprentissage supervisé , les données d'entraînement qui alimentent l'algorithme incluent la solution voulue qu'on appelle étiquette(Label).Une tâche typique de l'apprentissage supervisé est la classification, mais il y a aussi la régression [12].

- **la classification :** c'est une tache ou le modèle apprend a prédire la classe ou la catégorie d'une nouvelle données a partir des anciennes.

- **la régression :** c'est une méthode utilisé pour modéliser les relation entre les variable d'entrée et de sorties.

1.4.2 Apprentissage non supervisé

Comme on peut le deviner, l'apprentissage non supervisé n'utilise pas d'étiquettes, le programme n'a pas besoin de surveillance lors de son apprentissage.Parmi les algorithmes d'apprentissage non supervisé les plus importants, on retrouve :

- **Clustering :** c'est une famille d'algorithme qui efficient dans le partitionnement et le regroupement d'un ensemble de données (non étiquette) en différent groupe.
- **La détection d'anomalie :** ce sont des algorithme qui consistent a identifier des observation rare et inhabituel dans un jeux de donnée, ce sont les données qui diffèrent de la règle générale de son jeu.

1.4.3 Apprentissage par renforcement

L'apprentissage par renforcement est très différent de ce qu'on a vu jusqu'à maintenant.Le système d'apprentissage, appelé "agent" dans ce contexte, peut sélectionner et exécuter des actions basées sur ce qu'il a observé dans l'environnement, puis en retour recevoir des récompenses ou des pénalités suivant ces choix [12].C'est de la même façon qu'un enfant apprend dès ses premiers choix dans la vie.

Il doit alors apprendre par lui-même quelle est la meilleure stratégie, appelée politique dans le premier cas, une politique définit quelle action choisir lorsqu'il se trouve dans une situation donnée.

1.4.4 Apprentissage par transfert

L'apprentissage par transfert est une technique du machine learning où un modèle entraîné sur une tâche est réutilisé (en tout ou en partie) pour une autre tâche, souvent différente mais liée.

Cette méthode est très populaire dans le domaine du Deep learning car elle permet d'entraîner des réseaux de neurones profonds avec relativement peu de données [13].

1.5 Modèles de Machine Learning

Après avoir présenté les différentes catégories d'apprentissage en Machine Learning, il est important de s'intéresser aux modèles qui permettent de les mettre en œuvre . Chaque typologie repose sur des algorithmes spécifiques, adaptés à la nature des données et aux objectifs visés. Dans la section suivante, nous allons explorer quelques modèles de Machine Learning et leurs principes de fonctionnement.

1.5.1 Modèles d'apprentissage supervisé

1.5.1.1 Régression linéaire

La régression linéaire est l'un des modèles fondamentaux de l'apprentissage supervisé. On considère un vecteur d'entrée $X^T = (X_1, X_2, ..., X_p)$, et on cherche à prédire une variable de sortie réelle Y. Le modèle s'écrit sous la forme :

$$f(X) = \beta_0 + \sum_{j=1}^{p} X_j \beta_j$$

Le modèle suppose que la fonction de régression $E(Y \mid X)$ est linéaire, ou du moins que l'approximation linéaire est raisonnable. Les coefficients β_j sont inconnus et doivent être estimés à partir des données. Les variables explicatives X_j peuvent provenir de différentes sources :

- **Variables quantitatives** : par exemple l'âge, la taille, etc.
- **Transformations de variables** : log, racine carrée, carrés, etc.
- **Développements en base (polynômes)** : comme $X_2 = X_1^2$, $X_3 = X_1^3$, etc.
- **Interactions entre variables** : par exemple $X_3 = X_1 \cdot X_2$.

Quelle que soit la source des variables X_j, le modèle reste linéaire par rapport aux paramètres β_j.

Pour estimer les coefficients, on suppose disposer d'un ensemble d'apprentissage $\{(x_i, y_i)\}_{i=1}^{N}$, où $x_i = (x_{i1}, x_{i2}, ..., x_{ip})^T$ représente les caractéristiques de l'observation i, voir . La méthode des moindres carrés consiste à minimiser la somme des carrés des résidus (RSS) :

$$RSS(\beta) = \sum_{i=1}^{N} (y_i - f(x_i))^2 = \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2$$

On choisit alors les coefficients $\beta = (\beta_0, \beta_1, ..., \beta_p)^T$ qui minimisent cette erreur.

Tel que :

- y_i : la valeur observée de la variable dépendante (ou cible) pour l'observation i.
- $\hat{y}_i$: la valeur prédite de la variable dépendante y_i par le modèle.
- β_0 : l'ordonnée à l'origine, c'est-à-dire la constante du modèle.
- β_j : le coefficient de régression associé à la variable indépendante x_j.
- x_{ij} : la valeur de la variable explicative x_j pour l'observation i.
- p : le nombre total de variables explicatives (ou caractéristiques).

1.5.1.2 Random Forest

Les forêts aléatoires constituent l'une des idées d'apprentissage les plus puissantes introduites au cours des vingt dernières années.

L'idée essentielle des forêts aléatoires est d'améliorer la réduction de variance obtenue par le bagging en réduisant la corrélation entre les arbres, sans trop augmenter la variance individuelle de chaque arbre. Cela est réalisé pendant le processus de construction de l'arbre par une sélection aléatoire des variables d'entrée [15].

L'algorithme de la Forêt Aléatoire pour la régression ou la classification est le suivant :

1. Pour $b = 1$ à B (où B est le nombre d'arbres à construire) :

a. Tirer un échantillon bootstrap Z^* de taille N à partir des données d'entraînement.

b. Développer un arbre de forêt aléatoire T_b sur les données bootstrappées, en répétant récursivement les étapes suivantes pour chaque nœud terminal de l'arbre, jusqu'à ce que la taille minimale du nœud (n_{min}) soit atteinte :

   i. Sélectionner m variables au hasard parmi les p variables disponibles.

   ii. Choisir la meilleure variable/point de division parmi les m sélectionnées.

   iii. Diviser le nœud en deux nœuds filles.

2. Produire l'ensemble des arbres $\{Tb\}_1^B$

Pour faire une prédiction sur un nouveau point x :

- **Régression :** La prédiction agrégée est la moyenne des prédictions de chaque arbre : $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.
- **Classification :** Soit $\hat{C}_b(x)$ la prédiction de classe du b-ième arbre de forêt aléatoire. La prédiction finale est le vote majoritaire des classes prédites :

$$\hat{C}_{rf}^B(x) = \text{vote\_majoritaire}\left\{\hat{C}_b(x)\right\}_{b=1}^B$$

Typiquement, la valeur de m est choisie à $\lfloor\sqrt{p}\rfloor$ pour la classification et $\lfloor\frac{p}{3}\rfloor$ pour la régression, et la taille minimale du nœud (n_{min}) est de un pour la classification et cinq pour la régression. Ces valeurs par défaut peuvent être ajustées comme paramètres de réglage pour optimiser les performances [15].

Remarque 1. Le bagging, abréviation de "bootstrap aggregation", est une technique d'apprentissage automatique utilisée pour réduire la variance d'une fonction de prédiction estimée.

1.5.1.3 LSTM (Long Short-Term Memory)

Le Long Short-Term Memory (LSTM) est un type de réseau de neurones récurrents (RNN) spécialement conçu pour faire face aux problèmes des RNN classiques dans l'apprentissage des séquences longues, en particulier les problèmes de gradients qui disparaissent ou explosent lors de la rétropropagation du gradient à travers le temps.

Développé par Hochreiter et Schmidhuber en 1997 [17], le LSTM introduit un mécanisme de mémoire interne, contrôlé par des portes, permettant au modèle de conserver, d'oublier ou d'ajouter des informations à la mémoire de manière contrôlée.

Architecture d'un bloc LSTM : Un bloc LSTM repose sur une cellule de mémoire C_t qui stocke l'information et est régulée par trois portes principales :

- **La porte d'oubli (f_t) :** Elle détermine quelles informations de l'état précédent doivent être supprimées de la mémoire.
- **La porte d'entrée (i_t) :** Elle contrôle quelles nouvelles informations vont être ajoutées à la mémoire.
- **La porte de sortie (o_t) :**Elle détermine quelle partie de la mémoire interne sera transmise à la sortie.

Voici les équations associées à une cellule LSTM à l'instant t :

$$
\begin{aligned}
f_t &= \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) && \text{(Porte d'oubli)} \\
i_t &= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) && \text{(Porte d'entrée)} \\
\tilde{C}_t &= \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) && \text{(Candidat à la mémoire)} \\
C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t && \text{(Mise à jour de l'état de la cellule)} \\
o_t &= \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) && \text{(Porte de sortie)} \\
h_t &= o_t * \tanh(C_t) && \text{(État caché)}
\end{aligned}
$$

Avec :

- x_t : entrée à l'instant t,

- h_{t-1} : état caché précédent,
- C_{t-1} : mémoire précédente de la cellule,
- f_t, i_t, o_t : portes d'oubli, d'entrée et de sortie,
- $\tilde{C}_t$: nouveau contenu candidat de la mémoire,
- W et b : poids et biais à apprendre,
- σ : fonction sigmoïde,
- tanh : tangente hyperbolique,
- $*$: produit élément par élément.

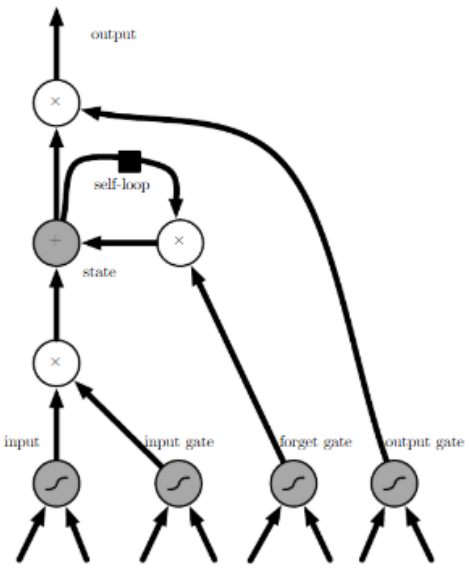

FIGURE 1.1 – schema explicatif des port d'un LSTM [13]

Remarque 2. Les problèmes de gradients qui disparaissent ou explosent sont des difficultés bien documentées dans l'apprentissage des réseaux neuronaux récurrents (RNN). Lors de la rétropropagation du gradient à travers le temps (Backpropagation Through Time), les gradients peuvent décroître exponentiellement (gradient qui disparaît) ou croître de manière incontrôlée (gradient qui explose), rendant l'apprentissage inefficace voire instable [4].

1.5.2 Modèles d'apprentissage non supervisé

1.5.2.1 K-means

L'algorithme K-means est une méthode de clustering non supervisé utilisée pour partitionner un ensemble de données en K groupes (ou clusters) homogènes. Il repose sur la minimisation de la somme des distances quadratiques entre chaque point et le centroïde de son cluster [22].

Chaque observation est affectée au cluster dont le centroïde est le plus proche, et les centroïdes sont mis à jour à chaque itération jusqu'à convergence.

$$\arg\min_{C} \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

où :

- C_k : ensemble des points affectés au cluster k,
- μ_k : centroïde du cluster k,
- $\|x_i - \mu_k\|^2$: distance euclidienne quadratique entre un point x_i et le centroïde μ_k.

Étapes principales de l'algorithme :

- **Etape 1 :** Initialiser K centroïdes $\mu_1, \mu_2, \ldots, \mu_K$
- **Etape 2 :**Affecter chaque point x_i au cluster dont le centroïde est le plus proche :

$$C_k = \{x_i : \|x_i - \mu_k\|^2 \leq \|x_i - \mu_j\|^2 \quad \forall j = 1, ..., K\}$$

- **Etape 3 :**Recalculer les centroïdes comme la moyenne des points dans chaque cluster :

$$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$

- **Etape 4 :**Répéter les étapes 2 et 3 jusqu'à convergence .

1.5.2.2 Autoencodeurs

Un autoencodeur est un type de réseau de neurones artificiels non supervisé conçu pour apprendre une représentation compressée des données en minimisant la perte d'information [16].

Il est constitué de deux parties symétriques :

1. un encodeur qui projette les données dans un espace latent(sortie de l'encodeur) de dimension réduite.
2. un décodeur qui reconstruit les données d'origine à partir de cette représentation compressée.

L'objectif est que la sortie du réseau soit la plus proche possible de l'entrée, ce qui permet notamment la réduction de dimension, la détection d'anomalies, ou la compression de données.

$$x \rightarrow [\text{Encodeur}] \rightarrow z \rightarrow [\text{Décodeur}] \rightarrow \hat{x}$$

Avec : x une entrée , z une représentation latente (codée) $\hat{x}$ une reconstruction de l'entrée.

les formules de l'encodeur et du décodeur :

$$z = f_\theta(x) = \sigma(W_e x + b_e)$$

$$\hat{x} = g_\phi(z) = \sigma(W_d z + b_d)$$

1.6 Apprentissage sur séries temporelles

1.6.1 Définition d'une série temporelle

Une série temporelle est une séquence d'observations numériques, ordonnées dans le temps.Elle peut être définie sur des intervalles réguliers ou irréguliers.Cette dernière sert à analyser ou à prévoir l'évolution d'un phénomène dans le temps [5].

On note généralement une série temporelle par :

$$Y_1, Y_2, Y_3 \ldots, Y_N$$

où chaque Y_t représente la valeur observée à l'instant t. la figure 1.2 présente les données du nombre de passagers aériens entre 1949 et 1960, il s'agit de la série temporelle `AirPassengers`, présente dans le package dataset chargé par défaut dans R.

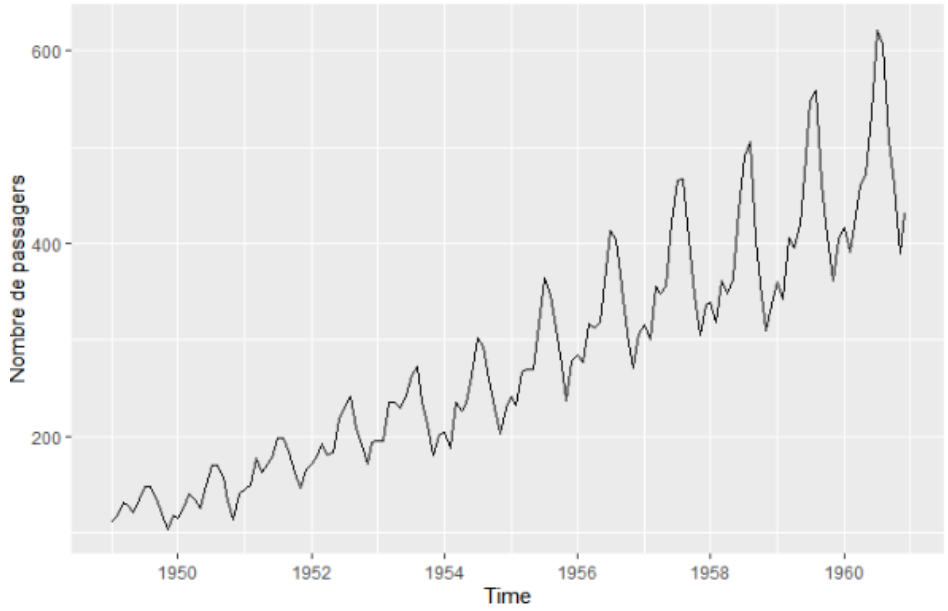

FIGURE 1.2 – Nombre de passagers aériens internationaux mensuels entre 1949 et 1960.

1.6.2 Les composantes d'une série temporelle

Une série chronologique a trois composantes : une tendance, une saisonnalité et une composante résiduelle, [18] .

- **La tendance** (T_t) : Elle représente l'évolution général de la série sur le long terme. Elle peut être croissante, décroissante ou constante. On dit qu'une tendance `Change de direction` lorsqu'elle passe d'une tendance à la hausse à une tendance à la baisse.
- **La saisonnalité** (S_t) : Elle désigne des motifs qui se répètent à intervalles réguliers dans le temps, comme les ventes mensuelles influencées par les saisons.
- **Résidu** (ε_t) : Elle représente les variations imprévisibles d'une série temporelle.

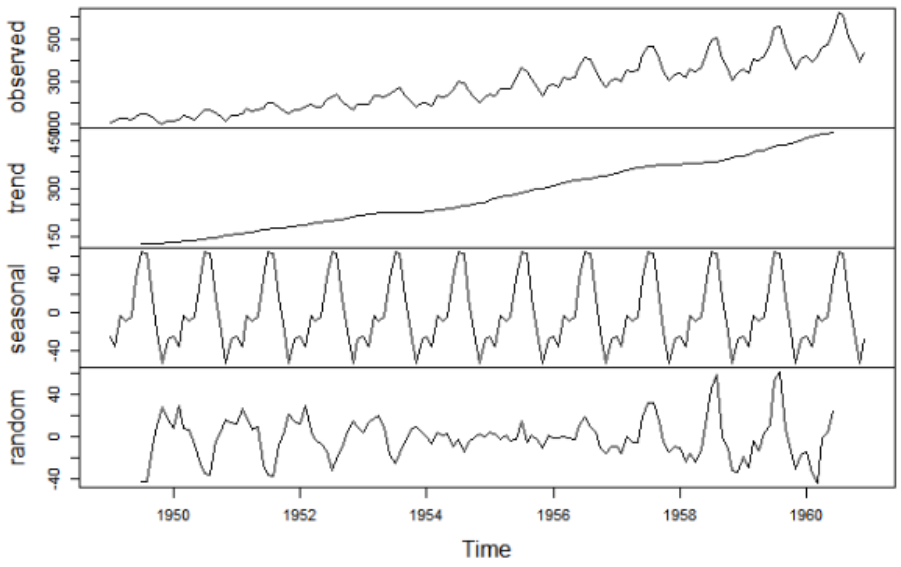

FIGURE 1.3 – décomposition de la série temporelle du nombre de voyageurs aériens mensuels.

1.6.3 Schémas d'une série temporelle

Modèle additif

Un modèle additif suppose que les différentes composantes d'une série temporelle (tendance, saisonnalité et bruit) s'additionnent, il est utilisé quand l'amplitude de la saisonnalité reste constante dans le temps(en d'autre terme les fluctuations saisonnières sont indépendantes du niveau de la série). Une observation Y_t est alors définie comme suit :

$$Y_t = T_t + S_t + \varepsilon_t$$

où :

- T_t représente tendance.
- S_t est la saisonnalité.
- ε_t est le bruit aléatoire ou résidu.

Modèle multiplicatif

Un modèle multiplicatif suppose que les composantes interagissent de manière proportionnelle, il est utilisé lorsque la variabilité saisonnière augmente ou diminue proportionnellement au niveau de la série.

Une observation Y_t est alors définie comme suit :

$$Y_t = T_t \times S_t \times \varepsilon_t$$

1.7 Modèles à base de lissage

Les méthodes de lissage constituent l'un des fondements historiques de l'analyse et de la prévision des séries temporelles. Elles reposent sur le principe d'une atténuation progressive des fluctuations aléatoires ou irrégulières en appliquant un lissage aux valeurs observées. Ce procédé permet de faire émerger plus distinctement les structures sous-jacentes de la série, notamment la tendance ou, dans certains cas, la saisonnalité. Comme le soulignent Hyndman et Athanasopoulos [18], ces techniques sont particulièrement efficaces pour produire des prévisions à court terme lorsque la structure de la série est stable et relativement peu bruitée.

1.7.1 Lissage par la moyenne mobile

Le lissage par la moyenne mobile représente l'une des approches les plus élémentaires pour lisser une série temporelle. Elle consiste à substituer chaque valeur observée par la moyenne arithmétique des n dernières observations.

$$\hat{y}_t = \frac{1}{n} \sum_{i=0}^{n-1} y_{t-i}$$

où :

- n est la taille de la fenêtre (par exemple 7 jours)
- y_t est la valeur observée à l'instant t
- $\hat{y}_t$ est la valeur lissée

1.7.2 Les lissages exponentiels

Les méthodes de lissage exponentiel constituent une approche simple mais efficace pour le lissage et la prévision des séries temporelles. Elles reposent sur le principe d'une pondération exponentiellement décroissante des observations

passées, accordant ainsi davantage d'importance aux données les plus récentes. Cette stratégie permet de capturer de manière réactive les évolutions récentes de la série tout en conservant une mémoire des comportements antérieurs.

Il existe plusieurs variantes du lissage exponentiel, adaptées à différents types de séries :

Lissage exponentiel simple

Le lissage exponentiel simple est une amélioration de la moyenne mobile, dans laquelle les pondérations décroissent de manière exponentielle avec le temps. Il est adapté aux séries temporelles sans tendance ni saisonnalité.

$$\hat{y}_{t+1} = \alpha y_t + (1-\alpha)\hat{y}_t$$

Tel que :

- $00 < \alpha < 1$:paramètre de lissage.
- y_t : valeur observée à l'instant t.
- $\hat{y}_t$: estimation du niveau à l'instant t.

Le paramètre α, compris entre 0 et 1, est appelé coefficient de lissage. Il détermine le poids accordé à la valeur la plus récente dans le calcul de la prévision. Une valeur de α proche de 1 rend le modèle plus sensible aux nouvelles observations, tandis qu'une valeur plus faible accorde davantage d'importance aux valeurs passées.

Lissage exponentiel double (Holt)

Le lissage exponentiel double, introduit par Charles Holt, est une extension du lissage exponentiel simple.cette méthode est particulièrement adaptée aux séries dépourvues de saisonnalité, mais affichant une croissance ou une décroissance régulière dans le temps,[18]. Les équations récurrentes de la méthode de Holt s'expriment comme suit :

$$\begin{cases} \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta) b_{t-1} \\ \hat{y}_{t+h} = \ell_t + h\, b_t \end{cases}$$

Tel que :

- $\ell_t$représente le niveau estimé à l'instant t.
- b_t désigne la tendance estimée à l'instant t.
- $\hat{y}_{t+h}$ est la prévision pour h périodes future.
- α et β sont des coefficients de lissage, chacun compris dans l'intervalle $(0,1)$.

Lissage exponentiel triple (Holt-Winters)

Le lissage exponentiel triple, également connu sous le nom de méthode de Holt-Winters, est une extension des méthodes de lissage précédentes permettant de modéliser des séries temporelles présentant à la fois une tendance et une composante saisonnière.
Cette approche repose sur la mise à jour simultanée de trois composantes : le niveau, la tendance, et la saisonnalité. Deux variantes principales existent : le modèle additif, adapté à une saisonnalité constante, et le modèle multiplicatif, mieux adapté aux séries dont l'amplitude saisonnière évolue proportionnellement au niveau général de la série,[7].

Les équations de la méthode de Holt-Winters pour le modèle additif :

$$\begin{cases} \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ s_t = \gamma(y_t - \ell_t) + (1-\gamma)s_{t-m} \\ \hat{y}_{t+h} = \ell_t + hb_t + s_{t+h-m(k+1)} \end{cases}$$

Formules du modèle multiplicatif :

$$\begin{cases} \ell_t = \alpha\left(\frac{y_t}{s_{t-m}}\right) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ s_t = \gamma\left(\frac{y_t}{\ell_t}\right) + (1-\gamma)s_{t-m} \\ \hat{y}_{t+h} = (\ell_t + hb_t) \cdot s_{t+h-m(k+1)} \end{cases}$$

Tel que :

- ℓ_t : niveau estimé de la série à l'instant t,
- b_t : tendance estimée à l'instant t,
- s_t : composante saisonnière à l'instant t,
- y_t : valeur observée de la série au temps t,
- m : période de la saisonnalité,
- $\alpha \in (0,1)$: coefficient de lissage pour le niveau,
- $\beta \in (0,1)$: coefficient de lissage pour la tendance,
- $\gamma \in (0,1)$: coefficient de lissage pour la saisonnalité,
- $\hat{y}_{t+h}$: prévision à h périodes dans le futur,
- $k = \left\lfloor \frac{h-1}{m} \right\rfloor$: nombre de cycles saisonniers complets dans l'horizon de prévision.

conclusion

Ce premier chapitre a permis d'établir les bases théoriques nécessaires à la compréhension des approches modernes de traitement des données. De la définition des types de données jusqu'à la présentation des principaux modèles d'apprentissage automatique, il constitue un socle conceptuel solide pour aborder les applications concrètes développées dans le reste de ce mémoire. En particulier, les notions relatives aux séries temporelles et aux modèles LSTM seront approfondies dans les chapitres suivants, où elles joueront un rôle central dans les stratégies de prévision mises en œuvre.

CHAPITRE 2

Interprétation des modèles de machine learning

Introduction

En machine learning, un modèle est considéré comme une "boîte noire" lorsqu'il fournit des prédictions sans que l'on puisse comprendre la démarche ou le principe sur lequel il s'appuie pour les générer. Cela est généralement dû à la complexité algorithmique du modèle, ce qui entrave son interprétation par les humains.

Ainsi, la capacité à interpréter correctement ces modèles ou leurs résultats est cruciale pour établir la confiance, comprendre les décisions qu'ils prennent, et vérifier si ces décisions respectent les standards éthiques. Par exemple, dans le domaine médical, il est indispensable de pouvoir justifier les décisions prises par ces modèles. Si un modèle prédit qu'un patient est atteint d'une maladie grave, il est essentiel de savoir quelles sont les variables ou bien les critères qui ont influencé cette décision, afin d'éviter des diagnostics erronés, dans un contexte où l'erreur peut coûter la vie à un individu.

Par ailleurs, il faut noter que certains modèles de machine learning ne comprennent pas véritablement pourquoi une entrée donnée doit recevoir une certaine étiquette, mais apprennent simplement que certaines caractéristiques sont

corrélées à une étiquette. Par exemple, si un modèle est entraîné sur un jeu de données où les seuls objets orange sont des ballons de basket, il pourrait apprendre à classifier tous les objets orange comme des ballons de basket, sans comprendre la véritable nature de l'objet. Ce modèle pourrait obtenir une grande précision, même sur des images de test, malgré le fait qu'il ne saisisse pas la différence qui fait réellement la différence [20] .

2.1 Notions générales

2.1.1 Définition de l'interprétabilité et Explicabilité en ML

L'interprétabilité reste un concept flou et ambigu, sans définition formelle précise. Les chercheurs diffèrent dans la définition de l'interprétabilité. D'après Tim Miller, "L'interprétabilité est le degré auquel un humain peut comprendre la cause d'une décision", [23]. Doshi-Velez et B. Kim définissent l'interprétabilité en apprentissage automatique comme "la capacité à expliquer ou à présenter de manière compréhensible pour un humain", [9]. En somme, on peut dire que l'interprétabilité d'un modèle de ML est fortement corrélée à la capacité des humains à comprendre les décisions et les prédictions produites par ce modèle.

Un autre terme courant dans la littérature est «l'explicabilité », donnant lieu à l'orientation de l'intelligence artificielle explicable (XAI), [39]. Doshi-Velez et Kim disent à propos de l'explicabilité : "Les explications sont... la monnaie avec laquelle nous échangeons des croyances.", [9]. Cette définition manque clairement de rigueur mathématique.

Selon la norme ISO 22989, l'explicabilité est la propriété d'un système d'IA à indiquer les facteurs importants influençant les résultats du système d'une manière compréhensible par les humains.

Dans [20], Lipton souligne la différence entre les questions auxquelles tentent de répondre les deux familles de techniques : l'interprétabilité soulève la question « Comment le modèle fonctionne-t-il ? », tandis que les méthodes d'explication cherchent à répondre à « Que peut encore me dire le modèle ? ».

2.1.2 Raisons et importance de l'interprétabilité et de l'explicabilité

Les systèmes de prise de décision automatisée ne sont pas largement acceptés. Les êtres humains veulent comprendre une décision, ou du moins obtenir une explication pour certaines décisions. Cela s'explique par le fait que les humains ne font pas confiance aveuglément. La confiance est donc l'un des aspects moteurs de l'explicabilité, [6]. D'autres motivations incluent la causalité, la transférabilité, la capacité à fournir de l'information, la prise de décision équitable et éthique, [20], la responsabilité et la possibilité d'ajuster les décisions.

- **Confiance** : La confiance est essentielle pour le déploiement d'un modèle prédictif. Comprendre et connaître les forces et faiblesses du modèle est une condition préalable à la confiance humaine, et donc à son adoption, [6]. En effet, les humains font plus facilement confiance à un système capable d'expliquer ses résultats qu'à une boîte noire.

- **Causalité** : Il est important de s'assurer que seules des relations causales sont retenues par le modèle. Dans [20], l'auteur affirme que les chercheurs estiment que l'interprétation d'un modèle d'apprentissage automatique peut permettre de générer des hypothèses testables par les scientifiques, contribuant ainsi au développement de la recherche.

- **Vie privée** : Lorsqu'il s'agit de traiter des données sensibles, comme des dossiers médicaux ou des informations provenant de compagnies d'assurance, il est impératif de garantir la confidentialité et la protection des données.

- **Équité et prise de décision éthique** : Connaître les raisons d'une décision est une nécessité sociétale, et ce droit pourrait devenir officiel pour les citoyens de l'Union européenne, [14]. Assurer des prédictions impartiales est crucial pour éviter des problèmes tels que les biais raciaux.

- **Ajustement du modèle** : L'explication d'un modèle d'apprentissage automatique est essentielle pour ajuster ses paramètres en s'appuyant sur les connaissances expertes du domaine. Selon [37], l'explicabilité peut même enseigner aux experts comment améliorer leur propre prise de décision.

2.1.3 Typologie des Approches d'Interprétation

L'interprétabilité peut s'appuyer sur différents aspects de l'apprentissage automatique. On peut se focaliser directement sur le modèle afin d'expliquer ses décisions ; dans ce cas, le modèle est dit intrinsèquement interprétable (ou modèle transparent). En revanche, si le modèle est considéré comme potentiellement complexe (ou une boîte noire), on peut en fournir une explication à l'aide d'une analyse post-hoc. Par ailleurs, l'interprétabilité peut également porter sur les données elles-mêmes, en cherchant à mieux comprendre la structure, les relations entre les variables et les distributions présentes dans le jeu de données.

- **Interprétation intrinsèque (modèles transparents)** : Un modèle est dit transparent lorsqu'il est conçu pour être lisible de bout en bout, et interprétable sans recours à des méthodes externes. Il s'agit typiquement de modèles simples, tels que les arbres de décision ou la régression linéaire. L'inconvénient principal réside dans le fait qu'imposer une structure trop simple peut limiter les performances prédictives sur des tâches complexes. En effet, [27] notent que l'interprétabilité par le modèle (model-based interpretability) peut parfois entraîner une moindre précision lorsque les relations sous-jacentes sont complexes.

- **Interprétation Post-hoc** : Les méthodes post-hoc sont des méthodes qui s'appliquent sur le modèle après la phase d'entraînement d'un modèle complexe (boîte noire). Les approches a posteriori (post-hoc) peuvent être différenciées selon le niveau d'analyse qu'elles adoptent. Certaines visent à expliquer le modèle dans son ensemble ; on parle alors d'une interprétation globale. À l'inverse, d'autres approches se concentrent sur l'expli-

cation d'une prédiction individuelle ; il s'agit d'une interprétation locale, voir [34].

Il est important de noter que, qu'elles soient intrinsèques ou post-hoc, les approches d'interprétation peuvent être classées selon deux niveaux d'analyse complémentaires :

- **L'interprétation globale** : L'interprétation globale, qu'elle soit intrinsèque ou post-hoc, cherche à offrir une compréhension du fonctionnement d'un modèle dans sa totalité. Pour l'interprétation globale intrinsèque, le modèle lui-même est conçu pour être transparent dès le départ sans besoin d'outils externes mais on peut renforcée l'interprétation globale intrinsèque en ajoutant des contraintes d'interprétabilité pendant l'entraînement, voir [10]. Cependant, atteindre une interprétation globale peut être très difficile pour les modèles complexes, car la quantité d'informations à assimiler peut dépasser la capacité de compréhension humaine, rendant impossible d'imaginer mentalement des relations dans des espaces de haute dimension, voir [25].
  L'interprétation globale post-hoc vise à extraire des règles générales, à estimer l'importance des variables d'entrée, ou à visualiser les relations apprises, permettant de construire une compréhension approximative mais utile de son comportement global.

- **L'interprétation locale** : Les modèles localement interprétables sont généralement obtenus en concevant des architectures de modèles mieux justifiées, capables d'expliquer pourquoi une décision spécifique a été prise. Contrairement aux modèles globalement interprétables, qui offrent un certain degré de transparence sur le fonctionnement interne du modèle, les modèles localement interprétables fournissent aux utilisateurs une justification compréhensible pour une prédiction particulière, voir [10]. Par exemple dans un arbre de décision et pour une instance donnée on peut suivre le chemin dans l'arbre pour savoir quelles caractéristiques ont conduit

à la prédiction.

Après avoir compris le modèle de manière globale, nous nous focalisons sur son comportement local afin de fournir des explications locales pour des prédictions individuelles. Les explications locales visent à identifier la contribution de chaque caractéristique d'entrée à une prédiction spécifique du modèle. Comme les méthodes locales attribuent généralement une décision du modèle à ses caractéristiques d'entrée, elles sont également appelées méthodes d'attribution, voir [10].

2.2 Méthodes d'Interprétation (Post-hoc)

2.2.1 Méthodes Basées sur les Perturbations

2.2.1.1 Analyse de sensibilité globale (GSA)

L'analyse de sensibilité globale est une approche qui vise à quantifier la façon dont les variations des variables d'entrée d'un modèle affectent sa sortie globale. Le paradigme général des méthodes d'analyse de sensibilité globale (GSA) consiste en deux phases, l'échantillonnage et l'analyse. Dans un premier temps, un ensemble d'échantillons est généré pour une variable ou un facteur X_i. Ensuite, le vecteur de sortie Y est produit à l'aide d'un modèle f sur l'ensemble des variables d'entrée :

$$Y = f(X_1, \ldots, X_p)$$

Enfin, l'impact de chaque variable est analysé et évalué, voir [32].

Les méthodes les plus récurrentes dans l'analyse de sensibilité globale sont la méthode de Sobol et la méthode de Morris.

Méthode de Sobol

La méthode de Sobol se base sur la décomposition de la variance de la sortie du modèle, sous l'hypothèse que les variables d'entrée sont indépendantes et non corrélées. Les équations (2.1) à (2.3) montrent la décomposition de la variance de la variable Y selon l'approche de Sobol :

$$V(Y) = \sum_{i=1}^{p} V_i + \sum_{1 \leq i < j \leq p} V_{ij} + \cdots + V_{1,\ldots,p} \tag{2.1}$$

$$V_i = V\left(\mathrm{E}(Y \mid X_i)\right) \tag{2.2}$$

$$V_{ij} = V\left(\mathrm{E}(Y \mid X_i, X_j)\right) - V_i - V_j \tag{2.3}$$

L'analyse de sensibilité de Sobol évalue l'impact de chaque paramètre d'entrée, aussi bien isolément qu'en interaction avec d'autres paramètres, ce qui permet d'estimer des indices de sensibilité d'ordre un (premier ordre), deux (second ordre), total et d'ordre supérieur.

Les équations (2.1) et (2.2) permettent de mesurer les variations de premier et second ordre, tandis que les équations (2.4) et (2.5) expriment les indices de sensibilité correspondants :

$$S_i = \frac{V_i}{V(Y)} \tag{2.4}$$

$$S_{ij} = \frac{V_{ij}}{V(Y)} \tag{2.5}$$

Enfin, l'indice de sensibilité total est défini comme la somme de toutes les contributions à la variance dans lesquelles intervient la variable X_i, ce qui inclut ses effets directs et ses interactions, voir [32].

$$S_{T_i} = \sum_{u \subseteq \{1,\ldots,p\}, i \in u} \frac{V_u}{V(Y)} \tag{2.6}$$

Méthode de Morris

l'idée de base de la méthode de morris est construite à base de calcul des effets élémentaires (EE) pour chaque variable d'entrée. Pour réaliser cela, on divise l'intervalle de variation de chaque variable en p niveaux, et on choisit un pas Δ, défini comme un multiple prédéterminé de $\frac{1}{p-1}$, tel que $X_i + \Delta \leq 1$, afin d'explorer l'espace des variables sous forme de grille.

L'effet élémentaire pour la variable X_i est défini comme suit :

$$EE_i = \frac{F(X_1, \ldots, X_{i-1}, X_i + \Delta, X_{i+1}, \ldots, X_k) - F(X_1, \ldots, X_k)}{\Delta_i}$$

Après avoir effectué plusieurs simulations pour X_i, on calcule la moyenne et l'écart-type des effets élémentaires, notés μ_i et σ_i respectivement :

$$\mu_i = \frac{1}{r} \sum_{j=1}^{r} EE_i^{(j)}$$

$$\sigma_i = \sqrt{\frac{1}{r} \sum_{j=1}^{r} \left(EE_i^{(j)} - \mu_i \right)^2}$$

Campolongo *et al.* ont proposé une version modifiée de μ, notée μ^*, en prenant la valeur absolue des effets élémentaires afin de compenser l'annulation des effets de signes opposés dans les modèles non monotones, voir [33].

Ainsi :

- Un μ_i^* élevé indique une forte influence de X_i sur la sortie ;

- Un σ_i élevé suggère une interaction importante avec d'autres variables ou un effet non linéaire, voir [26]

2.2.1.2 LIME (Local Interpretable Model-agnostic Explanations)

LIME, qui signifie Local Interpretable Model-agnostic Explanations (Explications Locales Interprétables Agnostiques au Modèle), est une technique d'explication conçue pour rendre les prédictions des modèles d'apprentissage automatique, souvent opaques, compréhensibles pour les humains.

L'objectif de LIME est de fournir une compréhension qualitative de la relation entre les composants d'une instance et la prédiction du modèle. . Il s'agit d'une approche "agnostique au modèle", ce qui signifie qu'elle peut expliquer les prédictions de n'importe quel classifieur ou régresseur, en traitant le modèle original comme une boîte noire, voir [30].

LIME (*Local Interpretable Model-agnostic Explanations*) repose sur l'utilisation de modèles interprétables locaux, appelés modèles substituts, afin d'expliquer la prédiction d'un modèle complexe pour une instance donnée x.

Mathématiquement, la recherche du meilleur modèle explicatif g pour une instance x s'exprime comme suit :

$$\text{explication}(x) = \arg\min_{g \in G} \mathscr{L}(f, g, \pi_x) + \Omega(g)$$

- f désigne le modèle complexe original (par exemple, un modèle XGBoost ou un réseau de neurones),
- g est le modèle explicatif local (souvent une régression linéaire simple),
- G est la famille des modèles explicatifs possibles (comme l'ensemble des modèles de régression linéaire),

- $\mathscr{L}(f, g, \pi_x)$ est une fonction de perte (par exemple, l'erreur quadratique moyenne) qui mesure à quel point les prédictions du modèle g sont proches de celles du modèle f, en se concentrant autour de l'instance x,
- π_x est une fonction de proximité qui pondère les observations générées autour de x : les points proches de x ont plus de poids dans l'apprentissage local,
- $\Omega(g)$ est une fonction de régularisation qui mesure la complexité du modèle explicatif (par exemple, en pénalisant les modèles utilisant trop de variables).

Dans la pratique, LIME n'optimise que la fonction de perte $\mathscr{L}$ pendant l'entraînement. La complexité du modèle explicatif $\Omega(g)$ est généralement contrôlée de manière indirecte : c'est à l'utilisateur de fixer une contrainte sur la complexité du modèle g, comme par exemple le nombre maximal de variables explicatives autorisées dans la régression linéaire.

Cette méthode permet ainsi d'obtenir une explication localement fidèle tout en restant interprétable, [30].

LIME peut être appliqué selon le type des données : images, textes, tabulaires, graphiques..etc ; Par exemple, pour une image, LIME peut identifier quelles parties (ou superpixels) influencent le plus la décision du modèle ; pour un texte, il peut montrer quels mots ont le plus de poids ; et pour des données tabulaires, il met en évidence les variables les plus déterminantes localement. Cette adaptabilité rend LIME particulièrement utile dans les contextes de machine learning et l'interpretrabilité.

Exemple 1. Pour mieux comprendre le fonctionnement de LIME, nous allons utiliser une vue d'ensemble visuelle des différentes étapes (voir la figure 2.1). Le premier graphique à gauche représente l'image originale. Le graphique suivant montre les contours de toutes les zones superpixelisées. Le troisième graphique illustre une image perturbée, c'est-à-dire une version modifiée de l'originale, où certaines régions (en noir) ont été masquées afin d'évaluer l'impact

de ces perturbations sur la prédiction du modèle. Enfin, le dernier graphique met en évidence les zones les plus importantes identifiées par LIME pour prédire la classe « toucan ».

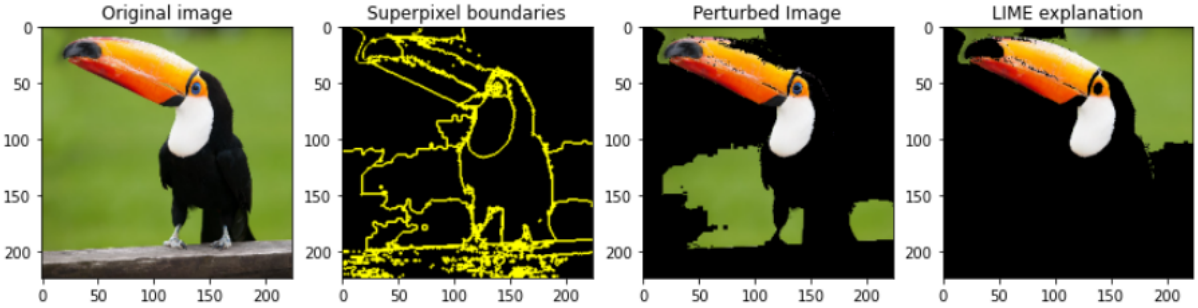

FIGURE 2.1 – Illustration des étapes de LIME appliquées à une image de toucan

2.2.2 Méthodes Basées sur l'Importance des Caractéristique

2.2.2.1 Feature Importance (Permutation)

L'importance des caractéristiques (Feature Importance, FI) est un outil fondamental pour comprendre comment les variables explicatives contribuent à la performance d'un modèle de prédiction. Elle mesure la dépendance du modèle vis-à-vis de chaque variable, en quantifiant à quel point la précision du modèle est affectée lorsqu'on perturbe les valeurs de cette variable,[11].

L'idée principale est la suivante : si l'on mélange aléatoirement les valeurs d'une caractéristique X_j (ce qui rompt son lien avec la variable cible y) et que cela entraîne une forte dégradation des performances du modèle, alors cette caractéristique est jugée importante,[25].

À l'inverse, si cette permutation n'affecte pas la performance du modèle, cela suggère que le modèle ne s'appuie pas sur cette caractéristique, la rendant ainsi peu importante,[25].

Une généralisation de cette notion a été proposée par Fisher, Rudin et Dominici sous le nom de *Model Reliance* (MR). Celle-ci est définie comme le rapport

entre :

$$MR_j = \frac{\mathbb{E}[\mathscr{L}(\hat{f}(X_{perm(j)}), y)]}{\mathbb{E}[\mathscr{L}(\hat{f}(X), y)]}$$

où $\mathscr{L}$ est une fonction de perte (comme l'erreur quadratique moyenne), $\hat{f}$ est le modèle, et $X_{perm(j)}$ est la matrice des données où la colonne j a été permutée,[11].

Une valeur de $MR_j > 1$ indique une dépendance du modèle à la caractéristique j, alors qu'une valeur proche de 1 suggère une absence de dépendance.

La méthode d'importance de permutation selon **Molnar .(2020)** [25], largement utilisée, suit les étapes suivantes :

1. **Évaluer l'erreur du modèle original :** Calculer l'erreur de prédiction $\mathscr{L}_{original} = \mathscr{L}(\hat{f}(X), y)$ sur les données d'origine.
2. **Perturber une caractéristique X_j :**
   - Permuter aléatoirement les valeurs de la colonne X_j pour obtenir une nouvelle matrice $X_{perm(j)}$.
   - Calculer l'erreur $\mathscr{L}_{perm(j)} = \mathscr{L}(\hat{f}(X_{perm(j)}), y)$ sur cette version permutée.
3. **Calculer l'importance de X_j :** On mesure l'importance par :

$$FI_j = \mathscr{L}_{perm(j)} - \mathscr{L}_{original} \quad \text{ou bien} \quad FI_j = \frac{\mathscr{L}_{perm(j)}}{\mathscr{L}_{original}}$$

4. **Trier les caractéristiques :** Enfin, les caractéristiques sont classées selon la valeur de FI_j, de la plus grande à la plus faible.

Cette méthode a l'avantage d'être *agnostique au modèle* (*model-agnostic*), car elle ne nécessite pas d'accéder à l'architecture interne du modèle $\hat{f}$: seules les prédictions sont utilisées.

Exemple 2. Comparons les deux variantes de l'importance des caractéristiques (*Feature Importance*) sur le jeu de données `mtcars`.

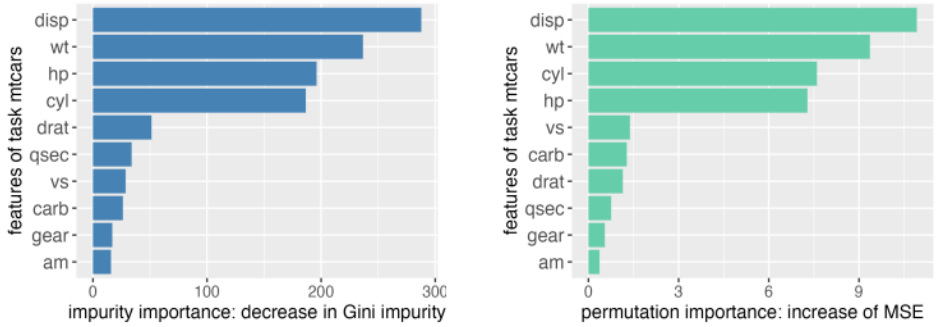

FIGURE 2.2 – Comparaison des méthodes d'importance des variables sur le jeu de données `mtcars` : importance par impureté (à gauche) et importance par permutation (à droite).

Les deux méthodes présentent un biais en faveur des variables ayant un plus grand nombre de niveaux, c'est-à-dire :

- les variables continues.
- ou les variables catégorielles avec de nombreuses catégories.

Ce problème a été mis en évidence par **Strobl et al. (2007)**.

Des versions plus avancées ont été développées pour corriger ces biais. En particulier, l'Importance par Permutation (PFI) et l'Importance des Caractéristiques (FI) ont été généralisées dans ce but.

2.2.2.2 SHAP (Shapley Additive explanations)

SHAP est une méthode locale pour expliquer les prédictions individuelles d'un modèle de *machine learning* en utilisant les valeurs de Shapley. Elle quantifie l'impact de chaque caractéristique sur la prédiction d'un modèle. SHAP s'accompagne de méthodes d'interprétation telles que *KernelSHAP*, une approche basée sur le noyau, et *TreeSHAP*, spécifique aux modèles d'arbres, [25].

SHAP décompose la prédiction de la manière suivante :

$$f(x) = \Phi_0 + \sum_i \Phi_i z_i,$$

où f est le modèle entraîné sur l'ensemble de données x, $x \in R^p$, $z_i \in \{0,1\}$ est le vecteur de coalition, et Φ_i la valeur de Shapley. La valeur de Shapley est

donnée par :

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \times (|N| - |S| - 1)!}{|N|!} \Big[f(S \cup \{i\})(x_{S \cup \{i\}}) - f(S)(x_S) \Big],$$

où S est une coalition de variables, $f(S \cup \{i\})$ le modèle entraîné sur le sous-ensemble de variables $S \cup \{i\}$, et $x_{S \cup \{i\}}$ le vecteur des valeurs de x restreint à ce sous-ensemble.

KernelSHAP approxime les valeurs de Shapley en utilisant une régression linéaire pondérée. Cette méthode consiste à générer de manière aléatoire des combinaisons de variables, puis à pondérer leur contribution en fonction de leur proximité avec l'observation d'intérêt. Elle est particulièrement utile lorsque le modèle est une « boîte noire » (comme certains modèles d'apprentissage non linéaire).

TreeSHAP est une méthode optimisée pour les modèles d'arbres (comme les forêts aléatoires ou les gradient boosting). Elle s'appuie sur la structure de l'arbre afin de calculer exactement les valeurs de Shapley de manière efficace, sans nécessiter d'approximation, [21]. Cette méthode consiste en fait à estimer l'espérance conditionnelle $E[f(x)|x_S]$ afin de quantifier la contribution de chaque variable :

$$\Phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \times (N - |S| - 1)!}{N!} \Big[E[f(x)|x_{S \cup \{i\}}] - E[f(x)|x_S] \Big],$$

où $E[f(x)|x_S]$ représente l'espérance conditionnelle de $f(x)$ donnée le sous-ensemble S.

Exemple 3. Un modèle de forêt aléatoire (Random Forest Regressor) a été entraîné pour prédire la valeur médiane des maisons en Californie, à partir des données disponibles dans le jeu de données California Housing proposé par la bibliothèque scikit-learn, [3].

Une fois le modèle entraîné, la méthode SHAP (SHapley Additive exPlanations) a été utilisée pour interpréter les contributions de chaque variable explicative aux prédictions réalisées par le modèle..

La figure suivante montre un graphique en barres représentant l'importance moyenne absolue des valeurs SHAP pour chaque variable explicative.

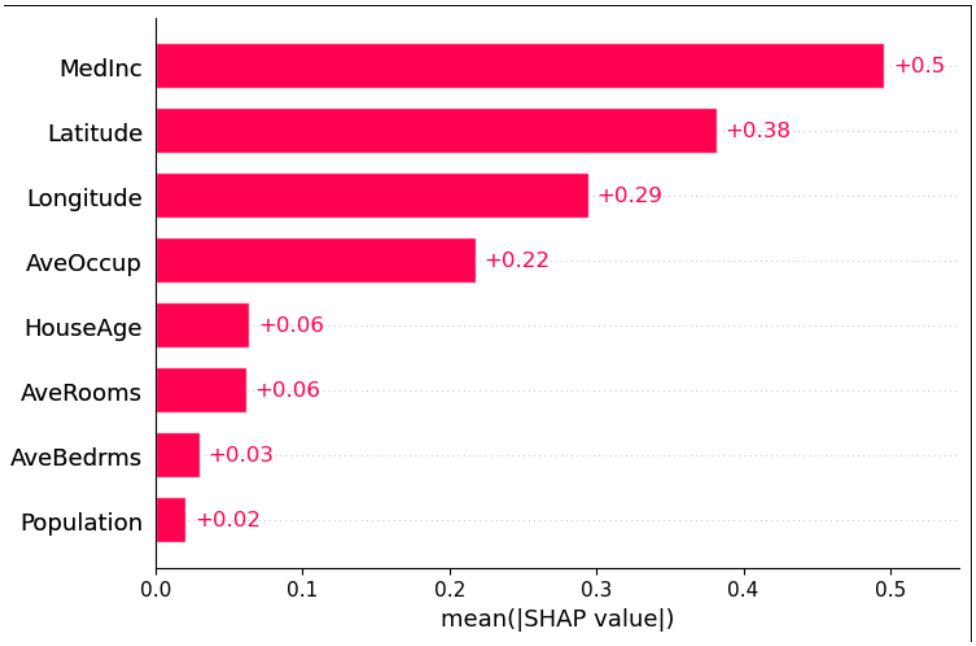

FIGURE 2.3 – graphique en barres représentant l'importance moyenne absolue des valeurs SHAP pour chaque variable explicative, [3].

Cette visualisation permet d'identifier l'importance relative des variables dans les prédictions du modèle. On observe que la variable MedInc (revenu médian) est la plus influente, suivie de Latitude et Longitude, ce qui suggère un fort lien entre les conditions géographiques et le phénomène prédit (probablement le prix de l'immobilier, si l'on se réfère au dataset californien souvent utilisé en exemple). Les variables comme Population ou AveBedrms ont une influence marginale sur la sortie du modèle.

2.2.3 Méthodes Basées sur la Visualisation

2.2.3.1 Saliency Map

Les Saliency Maps (ou cartes de saillance) représentent l'une des premières méthodes développées pour l'interprétation visuelle des réseaux de neurones convolutionnels (CNN). Proposées initialement par Simonyan et al. [38], elles visent à identifier les régions de l'image d'entrée qui ont le plus influencé la prédiction d'un modèle pour une classe donnée. Elles permettent ainsi de mieux comprendre le comportement du modèle en mettant en évidence les pixels ayant un impact significatif sur la sortie du réseau.

Principe de fonctionnement :

- Étant donnée une image d'entrée $\mathbf{X}$ et une classe cible c, on considère la fonction de score $S_c(\mathbf{X})$, représentant la sortie du modèle (avant softmax) pour la classe c. La carte de saillance est obtenue en calculant le gradient de ce score par rapport à l'image d'entrée :

$$\mathbf{M}_c = \left| \frac{\partial S_c(\mathbf{X})}{\partial \mathbf{X}} \right|.$$

  Ce gradient indique, pour chaque pixel, dans quelle mesure une variation locale affecte la probabilité de prédire la classe c. La valeur absolue permet de mettre en évidence l'intensité d'influence indépendamment du signe. .

- Ce gradient mesure la sensibilité de la prédiction vis-à-vis des variations de chaque pixel de l'image. Autrement dit, il indique dans quelle mesure une modification locale de l'image influerait sur la probabilité de la classe cible.

- En prenant la valeur absolue (ou parfois la valeur maximale par canal) de ce gradient, on obtient une carte de saillance qui peut être visualisée sous forme de carte thermique. Les pixels avec des gradients de forte intensité sont interprétés comme étant les plus déterminants dans la prise de décision du modèle.

Bien que cette méthode offre une visualisation à haute résolution de l'attention du modèle, elle présente certaines limitations. En particulier, elle est connue pour être sensible au bruit et peut produire des résultats peu interprétables lorsque l'image contient des objets multiples ou du contenu complexe [2]. Ces limites ont motivé le développement de méthodes alternatives plus robustes, telles que Grad-CAM ou SmoothGrad.

2.2.3.2 Grad-CAM (Gradient-weighted Class Activation Mapping)

Le Grad-CAM (Gradient-weighted Class Activation Mapping) est une technique proposée par Selvaraju et al. C'est une méthode de localisation discriminative de classe qui permet de visualiser les régions d'une image qui sont les

plus importantes pour la prédiction d'un concept cible spécifique par un modèle CNN, [36]. Au lieu de modifier l'architecture du réseau ou de nécessiter un réentraînement, le Grad-CAM exploite les gradients du score d'un concept cible (par exemple, la probabilité d'appartenir à une classe comme "chien", ou même une légende ou une réponse à une question). Ces gradients sont rétropropagés jusqu'à la dernière couche de convolution du CNN. La dernière couche de convolution est privilégiée car ses neurones conservent une information spatiale détaillée tout en capturant des concepts sémantiques de haut niveau, [36].

Le Mécanisme de fonctionnement de Grad-CAM est le suivant :

1. **Identification de la couche de convolution pertinente** : Le Grad-CAM utilise les informations de gradient provenant de la dernière couche de convolution du CNN. Cette couche est choisie car elle offre le meilleur compromis entre la conservation de l'information spatiale détaillée et la capture de concepts sémantiques de haut niveau, [36].

2. **Calcul des gradients du concept cible** : Pour un concept cible donné (par exemple, le score pour une classe spécifique comme "chien", ou même une légende ou une réponse à une question), le Grad-CAM calcule le gradient de ce score (y^c) par rapport aux cartes de caractéristiques (A^k) de la dernière couche de convolution. Cela permet de comprendre l'importance de chaque neurone de cette couche pour la décision d'intérêt.

3. **Calcul des poids d'importance des neurones** (α_k^c) : Ces gradients rétropropagés sont ensuite soumis à une moyenne globale spatiale (Global Average Pooling). Chaque valeur résultante, α_k^c, représente l'"importance" de la carte de caractéristiques k pour la classe cible c. Ce poids peut être interprété comme une linéarisation partielle du réseau profond en aval de la couche de convolution, [36].

La formule pour calculer ces poids est la suivante :

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

4. **Génération de la carte de localisation (Heatmap)** : Une fois les poids α_k^c obtenus, une combinaison pondérée des cartes de caractéristiques directes (A^k) est réalisée. Cette combinaison est ensuite passée à travers une fonction d'activation ReLU (Rectified Linear Unit). L'application de la fonction ReLU est cruciale car elle ne conserve que les activations qui ont une influence positive sur la prédiction de la classe d'intérêt. Les pixels ou régions ayant une influence négative (susceptibles d'appartenir à d'autres catégories dans l'image) sont ignorés, [36].

   La formule pour obtenir la carte de localisation brute $L_{Grad-CAM}^c$ est :

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

   Le résultat est une carte thermique grossière (heatmap), de la même taille que les cartes de caractéristiques de la couche de convolution, qui met en évidence les régions de l'image les plus importantes pour la prédiction du modèle concernant le concept ciblé. Les régions de forte intensité (souvent représentées en couleurs chaudes comme le rouge) indiquent les zones sur lesquelles le modèle s'est concentré, [28].

Exemple 4. Utilisant un modèle pré-entraîné VGG16 sur ImageNet. nous allons analyser une image d'un chien, et nous allons voir quelles zones de l'image activent le modèle pour reconnaître un chien.

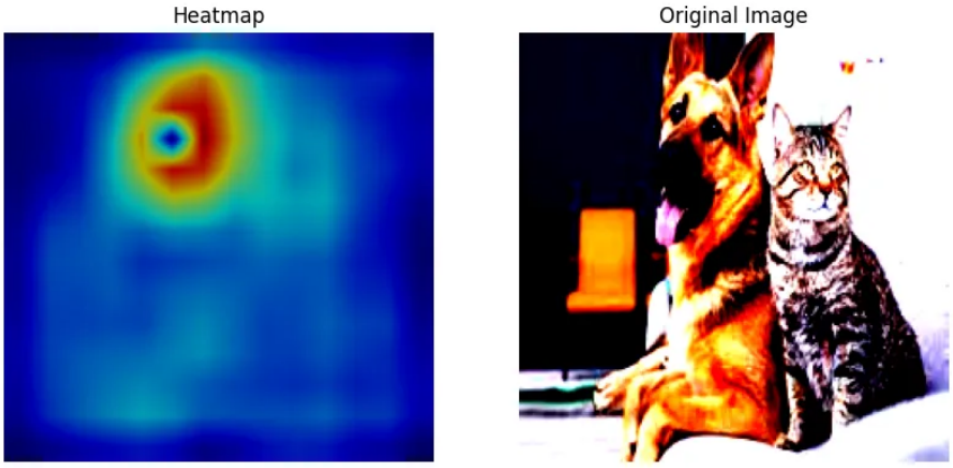

FIGURE 2.4 – Carte thermique des activations pour la classe "chien".

D'après l'interprétation obtenue par Grad-CAM, la carte thermique met en évidence que la zone d'activation principale se situe au niveau de la tête du chien, et plus précisément autour de son museau. Cela indique que le modèle concentre son attention sur cette région pour identifier la présence d'un chien dans l'image.

2.2.4 Analyse comparative des méthodes d'interprétation

Les méthodes d'interprétation utilisées dans ce travail peuvent être regroupées en trois grandes catégories : celles basées sur les perturbations, celles basées sur l'importance des caractéristiques, et celles reposant sur la visualisation. Chacune présente des avantages spécifiques, mais aussi certaines limitations. Le tableau ci-dessous en propose une synthèse comparative.

TABLE 2.1 – Avantages et inconvénients des méthodes d'interprétation utilisées

Méthode	Avantages	Inconvénients
GSA (Global Sensitivity Analysis)	Permet une compréhension globale du comportement du modèle face à différentes entrées. Utilisable pour des modèles complexes.	Coûteuse en calcul (nécessite de nombreuses simulations). Peu adaptée pour des explications locales.
LIME	Fournit une explication localement fidèle. Agnostique au modèle. Facile à interpréter avec des modèles simples (ex. : régression linéaire locale).	Sensibilité à la perturbation. Résultats non stables (aléatoire). Dépend du choix des paramètres et de la proximité définie.
Feature Importance (Permutation)	Simple à mettre en œuvre. Fournit une estimation de l'impact de chaque variable sur la performance du modèle.	Ne capture pas les effets d'interaction. Instable si les variables sont corrélées. Pas d'explication locale.
SHAP	Basé sur une théorie solide (valeurs de Shapley). Permet des explications locales et globales. Considère les interactions.	Coût computationnel élevé. Complexe à interpréter. Temps de calcul long pour les modèles lourds.
Saliency Map	Fournit une carte fine des pixels importants. Méthode simple basée sur les gradients.	Résultats bruités. Difficilement interprétable. Sensible aux variations.
Grad-CAM	Bonne lisibilité visuelle. Met en évidence des régions spatiales cohérentes. Moins sensible au bruit.	Résolution plus faible. Ne fonctionne qu'avec les couches convolutionnelles. Moins précise que les méthodes par pixel.

Conclusion

Ce chapitre a mis en lumière les enjeux majeurs liés à l'interprétation des modèles de machine learning, en soulignant le besoin croissant de rendre ces systèmes complexes plus compréhensibles et dignes de confiance.

Nous avons clarifié la distinction fondamentale entre interprétabilité et ex-

plicabilité, tout en justifiant leur rôle dans des domaines critiques où la transparence décisionnelle est essentielle. À travers une typologie structurée, le chapitre a exposé les différentes approches d'interprétation, intrinsèques ou post-hoc, locales ou globales, en illustrant leur mise en œuvre sur des modèles réels.

Les méthodes abordées GSA, LIME, permutation importance, SHAP, Saliency Maps et Grad-CAM ont chacune apporté un éclairage complémentaire sur la manière dont un modèle prend ses décisions. Leur diversité reflète la richesse des outils disponibles pour adapter l'analyse aux spécificités du problème traité, à la nature des données, et au niveau de granularité souhaité.

CHAPITRE 3

Application Numérique (cas d'étude BATELEC)

Introduction

Ce chapitre est consacré à la mise en œuvre concrète des techniques de science des données et d'apprentissage automatique dans un contexte industriel réel. Le cas d'étude sélectionné est celui de l'entreprise BATELEC, pour laquelle la prévision de la consommation de matières premières constitue un enjeu stratégique.

L'entreprise qui sert d'exemple pour l'étude est d'abord présentée, suivie par la problématique qu'elle pose. Ce chapitre se consacre au prétraitement et à l'anonymisation des données concernées. Aussi décrites sont les diverses étapes de l'analyse : application de modèles rangés sous le machine learning Random Forest et LSTM, interprétation des résultats obtenus à l'aide de méthodes d'interprétation (importance des variables, SHAP), et puis une évaluation comparative des performances.

3.1 Présentation de l'entreprise et la problématique

3.1.1 Présentation de l'entreprise

La SARL BATELEC, entreprise spécialisée dans l'électricité du bâtiment, a été fondée en 1987 dans la commune d'Ouzellaguen. Forte de son expérience,

elle s'est développée au fil des années pour devenir un acteur reconnu en Algérie, aussi bien dans la fabrication que dans la commercialisation de matériel électrique.

L'activité de commerce et de conseil est basée à Ouzellaguen. Une équipe dynamique et compétente y accueille les clients, les conseille et les accompagne dans leurs besoins. Des sociétés de renom telles que La Laiterie Soummam, Danone Djurdjura SPA, SARL Ibrahim et Fils, General Emballage, Ramdy, Groupe Batouche, GMF, Amimer Énergie, et bien d'autres, font déjà confiance à BATELEC.

L'activité de production se situe dans la zone d'activités de Taharacht, à Akbou. Elle est spécialisée dans la fabrication de gaines annelées et de boîtes de dérivation. Grâce à un personnel qualifié et dévoué, ainsi qu'à des équipements automatiques à forte capacité, l'entreprise garantit une production de qualité dans des délais maîtrisés. Un laboratoire interne permet également de tester les produits et d'assurer leur conformité aux exigences normatives.

3.1.2 Problématique

La société SARL BATELEC, spécialisée dans la distribution de matériaux électriques, fait face à des variations complexes et saisonnières de la consommation de ses produits, rendant difficile l'anticipation précise des besoins en stock. L'objectif est de développer des modèles de prévision de la consommation journalière à l'aide d'algorithmes de machine learning, capables de capter les dynamiques temporelles, les effets calendaires (jours fériés, saisons, etc.) ainsi que les comportements spécifiques à chaque article. Toutefois, au-delà de la performance prédictive, une compréhension fine du fonctionnement des modèles s'avère cruciale pour renforcer leur fiabilité et favoriser leur adoption en contexte industriel.

3.2 Collecte, préparation et anonymisation des données

3.2.1 La Collecte

Dans le cadre de notre étude, on a utilisé des données fournies par le responsable de la planification de l'entreprise BATELEC. Le jeu de données initial couvrait la période de janvier 2021 à mai 2025, avec un enregistrement quasi quotidien de la consommation de plusieurs matières premières (nommées articles dans le dataset).

Le jeu de données d'origine incluait des observations à partir de l'année 2021. Toutefois, cette première année a été volontairement supprimée de l'analyse. En effet, la pandémie de COVID-19 a eu un impact significatif sur le rythme de la production, affectant la consommation de la matière première au niveau de l'usine ; cela a créé des anomalies qui risquaient de fausser les prédictions futures.

	A	B	C
1	Date	Article	Quantité
2	2021-01-11 12:24:32	POLY-b	1700
3	2021-01-11 12:24:32	ADD+	75
4	2021-01-11 12:24:32	Fil de fer	1039.2
5	2021-01-14 08:41:00	ADD+	50
6	2021-01-14 15:17:01	POLY-b	513
7	2021-01-20 14:26:41	POLY-c	6200
8	2021-01-25 12:04:35	POLY-b	600
9	2021-01-25 15:31:47	POLY-c	200

FIGURE 3.1 – Le jeu de donnée brute

3.2.2 Le Prétraitement

Pour rendre les données exploitables par les modèles de machine learning, un prétraitement du jeu de données est indispensable. Nous avons donc procédé comme suit :

1. **Traitement des données manquantes** : Dans notre jeu de données, certaines observations comportaient des valeurs manquantes (quantité non déclarée), pour évaluer l'ampleur de ce phénomène nous avons calculé la proportion d'observations manquantes dans la colonne quantité. L'analyse a révélé que environs 4% des données étaient manquantes, un taux relativement faible mais non négligeable. Ces absences étaient dues à des pertes ponctuelles d'informations lors de l'enregistrement de la consommation quotidienne, rendant certaines lignes incomplètes. Afin de préserver la continuité temporelle, ces valeurs ont été modifiées à l'aide de la moyenne de la variable correspondante (quantité de l'article). Cette méthode a été choisie pour limiter les biais potentiels liés à la suppression d'observations, tout en conservant la structure globale du jeu de données.

2. **Conversion des dates** : les dates ont été sauvegardées sous forme de chaînes de caractères ; nous avons donc dû les convertir au format `datetime` dans Python.

3. **Création de variables temporelles dérivées** : à partir de la variable date, plusieurs attributs ont été générés pour enrichir notre jeu de données et faciliter l'apprentissage :
   - Le jour du mois (`jour_mois`)
   - Le jour de la semaine (`jour_semaine`)
   - Le mois (`mois`)
   - La saison (`saison`)
   - Le quartile de l'année (`quartile_annee`)
   - Le caractère férié ou non (`jour_ferie`)

   Ces variables temporelles ont été extraites dans le but de capturer des composantes saisonnières, hebdomadaires et calendaires, qui peuvent influencer les comportements de consommation.

4. **Encodage des variables** :

Les variables qualitatives ont été transformées en variables numériques afin de les rendre exploitables par les modèles de machine learning utilisés. Contrairement à un encodage *one-hot* classique, un encodage cyclique a été appliqué aux variables à caractère périodique telles que le *jour de la semaine*, le *jour du mois*, le *mois* ou encore le *quartile de l'année*.
L'encodage cyclique consiste à projeter chaque modalité sur un cercle trigonométrique à l'aide des fonctions sinus et cosinus, afin de préserver la continuité naturelle entre les extrémités du cycle. Par exemple, pour la variable `jour_semaine` (variant de 1 à 7), deux nouvelles variables sont générées selon les formules suivantes :

$$\text{jour\_semaine\_sin} = \sin\left(2\pi \cdot \frac{\text{jour\_semaine}}{7}\right)$$

$$\text{jour\_semaine\_cos} = \cos\left(2\pi \cdot \frac{\text{jour\_semaine}}{7}\right)$$

Ce type d'encodage permet au modèle de capturer la nature cyclique des données, en considérant par exemple que le dimanche est proche du lundi.

Remarque : Ainsi, l'encodage cyclique offre une meilleure représentation des dépendances périodiques dans le cadre des séries temporelles. Toutefois, l'interprétation des contributions individuelles des variables (feature importance) peut s'avérer moins intuitive, dans la mesure où chaque caractéristique cyclique est représentée par deux composantes (sin et cos), qu'il est alors indispensable d'agréger pour une lecture plus cohérente.

5. **Segmentation temporelle des données** : pour s'assurer que les modèles ne s'entraînent pas sur des données futures et pour respecter aussi la nature séquentielle des données :
   - L'ensemble d'entraînement contient les données allant de janvier 2022 jusqu'au 31 décembre 2024 (ce sont X_train et Y_train).
   - L'ensemble de test est constitué des données allant du 1er janvier 2025 jusqu'au 31 mai 2025 (ce sont X_test et Y_test).

3.2.3 Anonymisation

Aucune donnée personnelle identifiable n'était directement présente. Par mesure de précaution :

- Les noms d'articles (polymère, additif, fil de fer...) ont été remplacés par des identifiants anonymes (`POLY-a`, `ADD+`, `FIL...`).
- La quantité a été exprimée en unités anonymisées, sans faire référence à une mesure physique précise.

3.3 Application et interprétation des modèles de machine learning

3.3.1 Random Forest

Le modèle Random Forest est un algorithme d'apprentissage supervisé basé sur un ensemble d'arbres de décision. Il est efficace pour les tâches de régression et de classification, lorsque les relations entre les variables sont complexes ou non linéaires. Sa robustesse face au bruit et sa capacité à gérer des variables qualitatives et quantitatives en font un bon candidat pour notre étude (Le fonctionnement détaillé de l'algorithme est présenté au 1^{er} chapitre.).

le modèle Random Forest dans sa version régression (RandomForestRegressor), été utilisée pour effectuer des prédictions sur la consommation future des articles donnés, en exploitant à la fois les variables temporelles dérivées(le jour du mois, le mois ou la saison) et les indicateurs contextuels (les jours fériés). Un léger travail de feature engineering a également été réalisé afin d'enrichir l'information disponible, par l'ajout de variables de type retard (lag_1, lag_7) et d'une moyenne glissante sur 7 jours (rolling_mean_7). Ces variables permettent de mieux capturer l'inertie et la saisonnalité locale inhérentes aux séries temporelles (mieux expliquée dans la partie suivante).

3.3.1.1 Préparation des données

Le jeu de données est d'abord filtré pour ne conserver que l'article d'intérêt (cible). Ensuite, un encodage cyclique a été appliqué aux variables temporelles afin de les rendre exploitables par le modèle.

les variables utilisées par ce modèle sont jour_semaine, jour_mois, mois, quartile_annee, saison, jour_ferie, lag_1, lag_7 et rolling_mean_7.

Remarque 3. Les variables lag_1 et lag_7 correspondent respectivement à la quantité consommée la veille (J–1) et celle consommée une semaine auparavant (J–7). Les variables ont été introduites sur la base d'une analyse des autocorrélations. L'autocorrélogramme (ACF) a révélé des corrélations significatives aux retards 1 et 7, indiquant une dépendance immédiate (lag 1) et une récurrence hebdomadaire (lag 7) dans les données de consommation. Cette observation est confirmée par l'autocorrélogramme partiel (PACF), qui met en évidence une influence directe des mêmes lags sur la valeur actuelle. En complément, la variable rolling_mean_7, représentant la moyenne glissante sur 7 jours, permet de capturer la tendance locale en lissant les variations journalières. Ainsi, l'intégration de ces variables est statistiquement fondée et contribue à une meilleure modélisation de la dynamique temporelle de la série.
Les trois indicateurs lag_1, lag_2 et rolling_mean_7 ont donc été choisis pour enrichir le jeu de données avec des dynamiques temporelles simples, tout en restant compatibles avec l'architecture statique du modèle Random Forest.

voici la représentation graphique de l'analyse des autocorrélations effectuée sur tous les articles :

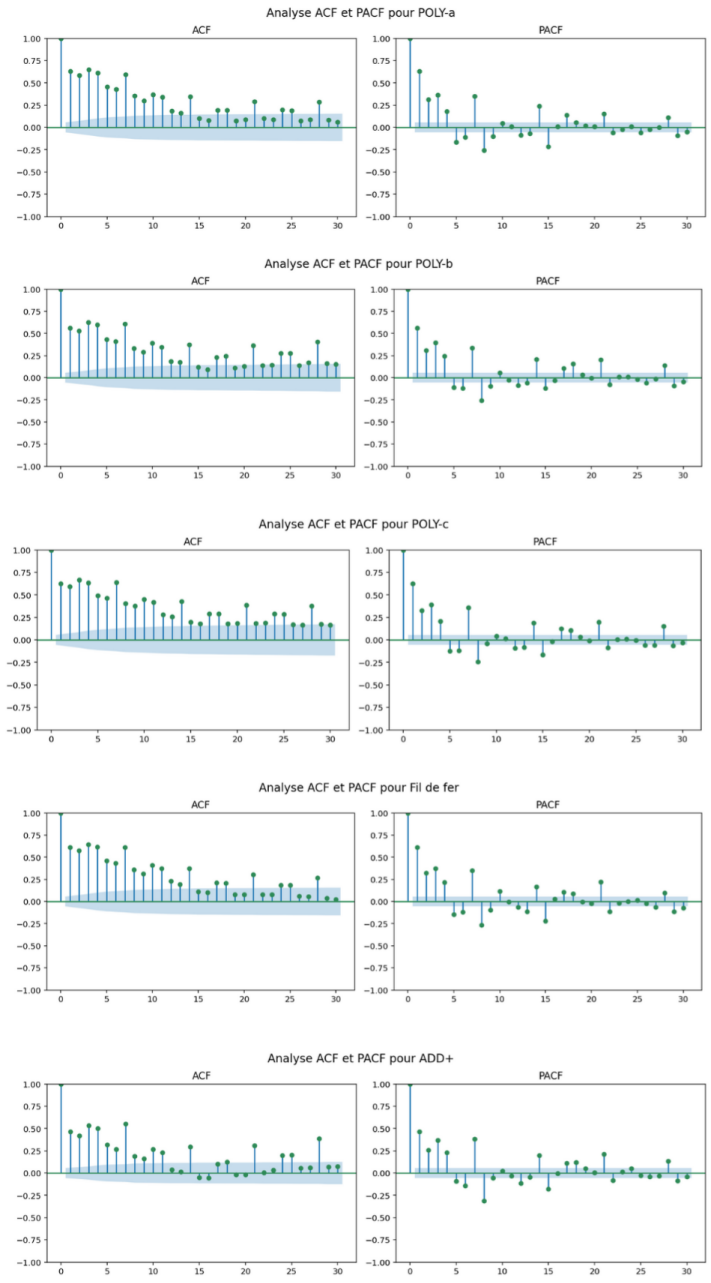

FIGURE 3.2 – Analyse des Autocorrelations AFC et PAFC

3.3.1.2 Modélisation et optimisation des paramètres

Le modèle Random Forest Regressor a été entraîné sur les données d'apprentissage. On utilise la validation croisée par GridSearchCV afin de déterminer la meilleure combinaison de paramètres, utilisant l'erreur quadratique moyenne négative pour l'évaluation avec une validation croisée à 5 plis. Les hyperparamètres optimisés sont :

- le nombre d'estimateurs (n_estimators).
- la profondeur maximale (max_depth).
- le nombre minimal d'échantillons pour diviser un nœud (min_samples_split).
- le nombre minimal d'échantillons par feuille (min_samples_leaf).

Les paramètres optimaux pour ce modèle pour tous les articles sont :

- le nombre d'estimateurs : 500.
- la profondeur maximale : None.
- le nombre minimal d'échantillons pour diviser un nœud : 10.
- le nombre minimal d'échantillons par feuille : 2.

L'entraînement du modèle RandomForestRegressor est réalisé avec les paramètres optimaux obtenus avec GridSearchCV et un Random State équivalent à 42 pour assurer la reproductibilité des résultats.

3.3.1.3 Résultats et évaluation

Le modèle a été évalué sur les données de l'année 2025, et ce sur cinq articles distincts : POLY-a, POLY-b,POLY-c,ADD+,fil de fer.Les métriques obtenues sont :

Article	**MSE**	**RMSE**	**R^2**
POLY-a	502.31	22.41	0.729
POLY-b	3026	55	0.755
POLY-c	2577.38	50.76	0.684
ADD+	7.80	2.79	0.764
fil de fer	103.38	10.16	0.834

TABLE 3.1 – Performance du modèle RandomForestRegressor sur les articles testés (année 2025)

Remarque : Le neg MSE (le MSE en valeur négative) est utilisé par GridSearchCV car GridSearchCV maximise les scores.

Afin de mieux visualiser les tendances générales et la saisonnalité sans être perturbé par la variabilité quotidienne, les courbes suivantes ont été lissées à l'aide d'une moyenne glissante sur 7 jours. Ce lissage n'affecte en rien la nature du modèle ni celle des prédictions, qui restent effectuées à une granularité journalière.

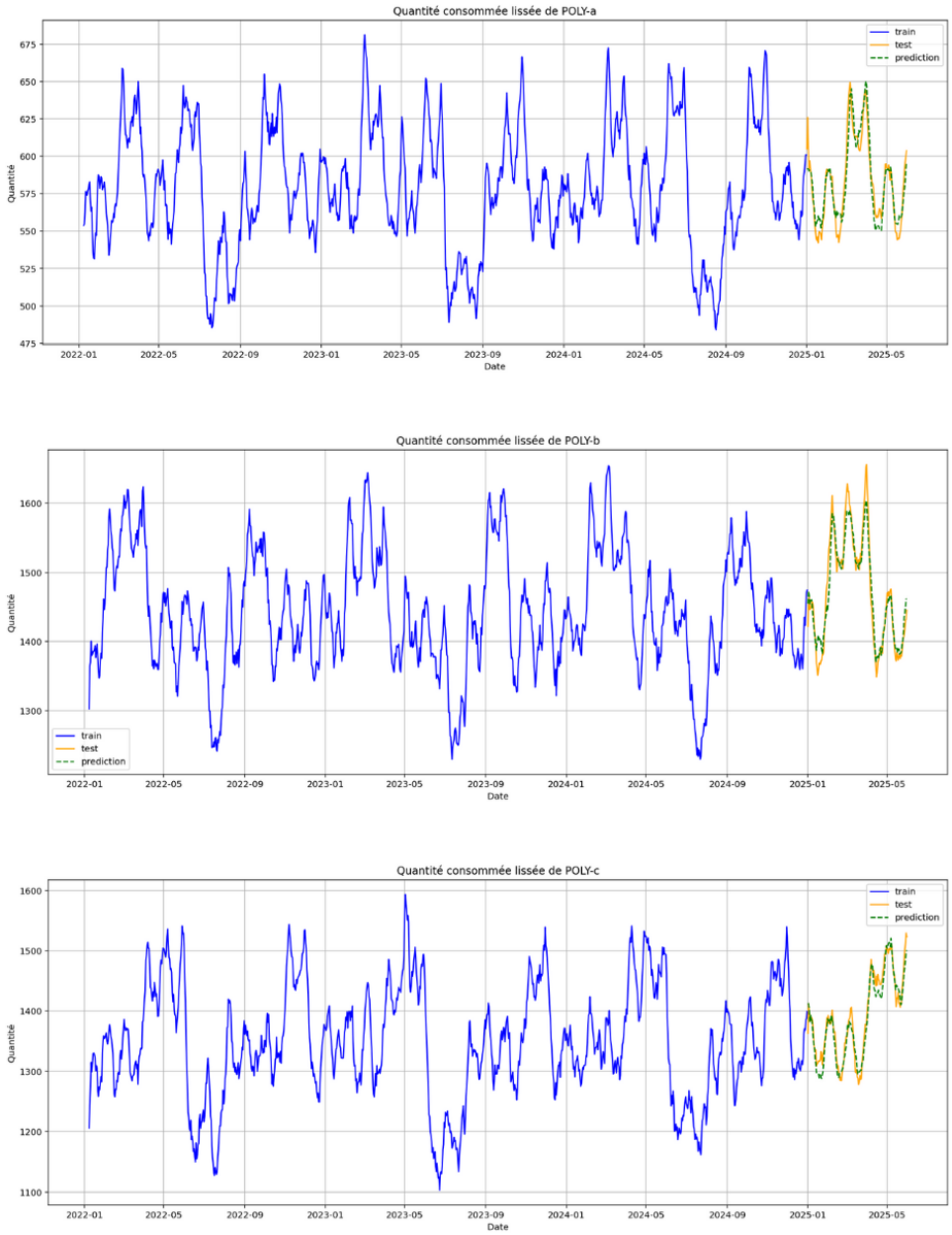

FIGURE 3.3 – Graphe comparatif entre les données réel et prédites en utilisant le modèle RandomForestRegressor (partie 1).

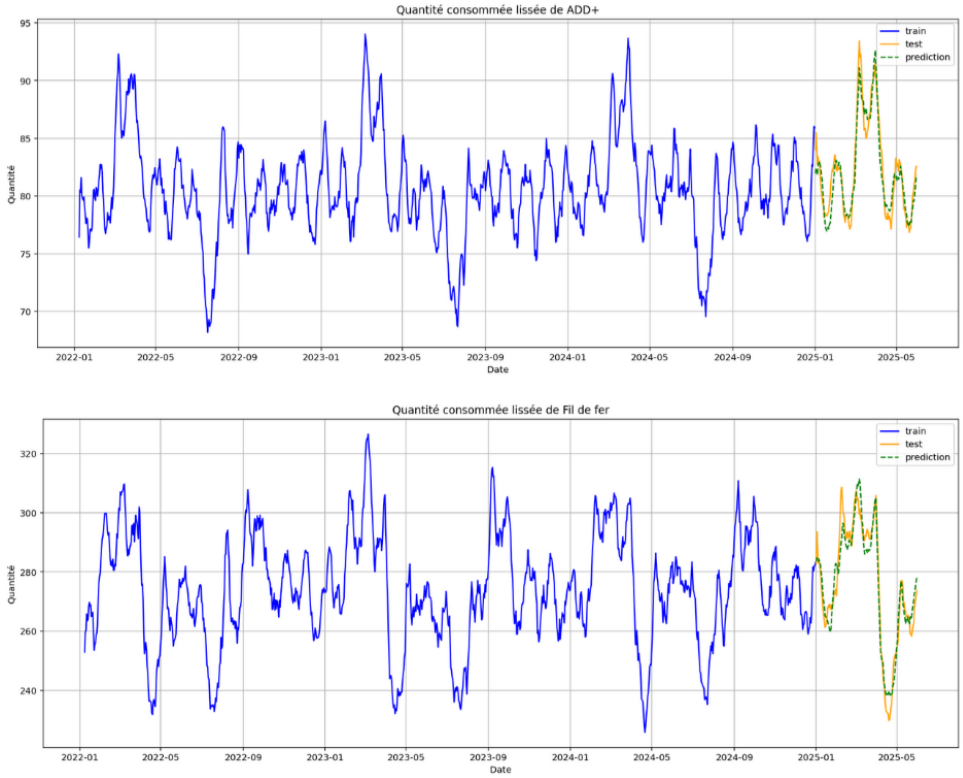

FIGURE 3.4 – Graphe comparatif entre les données réel et prédites en utilisant le modèle RandomForestRegressor (partie 2).

3.3.1.4 Analyse Générale de la Performance du Modèle

Le tableau 3.1 présente les métriques d'évaluation (MSE, RMSE, R^2) pour cinq articles distincts : POLY-a, POLY-b, POLY-c, ADD+, et Fil de fer, sur les données de l'année 2025.

R^2 (Coefficient de Détermination) : Cette métrique indique la proportion de la variance de la variable dépendante qui peut être prédite à partir des variables indépendantes. Des valeurs de R^2 plus élevées (plus proches de 1) sont préférables.

- L'article **Fil de fer** présente le R^2 le plus élevé à 0,834, ce qui suggère que le modèle explique une grande proportion de sa variance. Cela indique un bon pouvoir prédictif pour cet article.
- **ADD+** affiche également un R^2 élevé de 0,764.
- **POLY-b** et **POLY-a** obtiennent respectivement des valeurs de R^2 de 0,755 et 0,729, ce qui reste acceptable.

- **POLY-c** a le R^2 le plus faible à 0,684, ce qui signifie que le modèle explique une moindre part de sa variance comparé aux autres articles.

MSE (Erreur Quadratique Moyenne) et RMSE (Racine Carrée de l'Erreur Quadratique Moyenne) : Ces métriques quantifient l'ampleur moyenne des erreurs. Des valeurs plus faibles sont préférables.

- L'article **ADD+** présente des valeurs de MSE (7,80) et RMSE (2,79) remarquablement basses, en cohérence avec son R^2 élevé. Cela indique des prédictions très précises.
- **Fil de fer** obtient également des valeurs relativement basses (MSE = 103,38 ; RMSE = 10,16), ce qui confirme les bons résultats obtenus selon R^2.
- Les articles de la gamme **POLY** affichent des valeurs de MSE et RMSE significativement plus élevées. Par exemple, **POLY-b** enregistre un MSE de 3026 et un RMSE de 55, ce qui est attendu compte tenu des échelles plus importantes des quantités prédites. Ces métriques doivent donc être interprétées en tenant compte de la plage de variation des valeurs cibles.

En résumé, d'après les résultats du Tableau 3.1, le modèle **Random Forest Regressor** obtient les meilleures performances prédictives pour les articles **Fil de fer** et **ADD+**, suivis de **POLY-b**, **POLY-a**, puis **POLY-c**.

Les graphiques 3.5 et 3.6 comparent les quantités réelles (train/test) et prédites au fil du temps pour chaque article.

Tendance Générale : Pour tous les articles, le modèle capture généralement les tendances globales et la saisonnalité (hauts et bas) des quantités consommées. La ligne de prédiction suit largement la ligne de test.

Précision dans la Fenêtre de Prédiction (Janvier à Mai 2025) :

- **ADD+ et Fil de fer** : Comme prévu d'après leurs fortes valeurs de R^2, les prédictions pour ADD+ et le Fil de fer semblent s'aligner étroitement avec les données réelles de test pendant la période de prédiction 2025. Les lignes prédites (vertes en pointillés) sont très proches des lignes de test réelles (orange en pointillés). Cette observation visuelle confirme la haute précision rapportée dans le Tableau 3.1.

- **POLY-a, POLY-b, POLY-c** : Bien que le modèle capture le modèle général, il y a des écarts plus notables entre les valeurs prédites et réelles pour les articles POLY par rapport à ADD+ et Fil de fer, en particulier en termes d'amplitude des pics et des creux. Par exemple, le modèle peut légèrement surestimer certains pics ou sous-estimer certains creux. Cela correspond à leurs valeurs de R^2 relativement plus basses et à leurs métriques d'erreur plus élevées par rapport à ADD+.

Volatilité : Les données pour tous les articles montrent une volatilité quotidienne/hebdomadaire significative. Le modèle semble gérer cette volatilité raisonnablement bien, bien qu'il y ait des cas où les fluctuations rapides ne sont pas parfaitement reproduites par les prédictions.

L'analyse visuelle confirme que le modèle est plus performant pour ADD+ et le Fil de fer en termes de suivi précis des valeurs réelles dans la période de prédiction, ce qui est cohérent avec les métriques quantitatives.

3.3.2 Importance des variables explicatives

Après l'entraînement du modèle sur les données de consommation des différents articles, nous avons extrait les importances relatives de chaque variable explicative.

Cette importance est calculée en mesurant à quel point chaque variable contribue à la réduction de l'erreur au niveau des arbres de décision. Les variables qui permettent une meilleure séparation des données reçoivent une importance plus élevée.

Dans notre cas, les variables explicatives incluaient des attributs temporels encodés de manière cyclique (*mois, jour du mois, jour de la semaine, quartile de l'année, saison*), une variable binaire (*jour férié*) ainsi que des variables de mémoire (*valeurs décalées et moyenne glissante*) pour intégrer l'influence de la consommation passée.

Pour améliorer la lisibilité, les importances ont été agrégées selon des groupes de variables issus de la même origine. Par exemple, les composantes sin et cos du mois ont été regroupées sous une seule importance "mois".

Les résultats de la fonction *feature_importance* sont les suivants :

Feature	POLY-a	POLY-b	POLY-c	ADD+	Fil de fer
mois	0.0506	0.0704	0.2498	0.3362	0.1167
jour_mois	0.1612	0.1742	0.1409	0.2652	0.1381
jour_semaine	0.0000	0.0000	0.0000	0.0000	0.0000
saison	0.0000	0.0000	0.0000	0.0000	0.0000
quartile_annee	0.0000	0.0000	0.0000	0.0000	0.0000
jour_ferie	0.0011	0.0008	0.0007	0.0011	0.0009
lag_1	0.0897	0.0809	0.0621	0.0662	0.0712
lag_7	0.1056	0.1463	0.1052	0.1593	0.1003
rolling_mean_7	0.0000	0.0000	0.0000	0.0000	0.0000

TABLE 3.2 – Importance des groupes de variables par article

Les résultats du tableau ci-dessus indiquent que les features les plus importantes dans les prédictions sont :

- **POLY-a** : jour_mois, lag_7, lag_1.
- **POLY-b** : jour_mois, lag_7, lag_1.
- **POLY-c** : mois, jour_mois, lag_7.
- **ADD+** : mois, jour_mois, lag_7.
- **Fil de fer** : jour_mois, mois, lag_7.

Les features les moins importantes dans les prédictions sont :

- **POLY-a** : jour_semaine, saison, quartile_annee, rolling_mean_7.
- **POLY-b** : jour_semaine, saison, quartile_annee, rolling_mean_7.
- **POLY-c** : jour_semaine, saison, quartile_annee, rolling_mean_7.
- **ADD+** : jour_semaine, saison, quartile_annee, rolling_mean_7.
- **Fil de fer** : jour_semaine, saison, quartile_annee, rolling_mean_7.

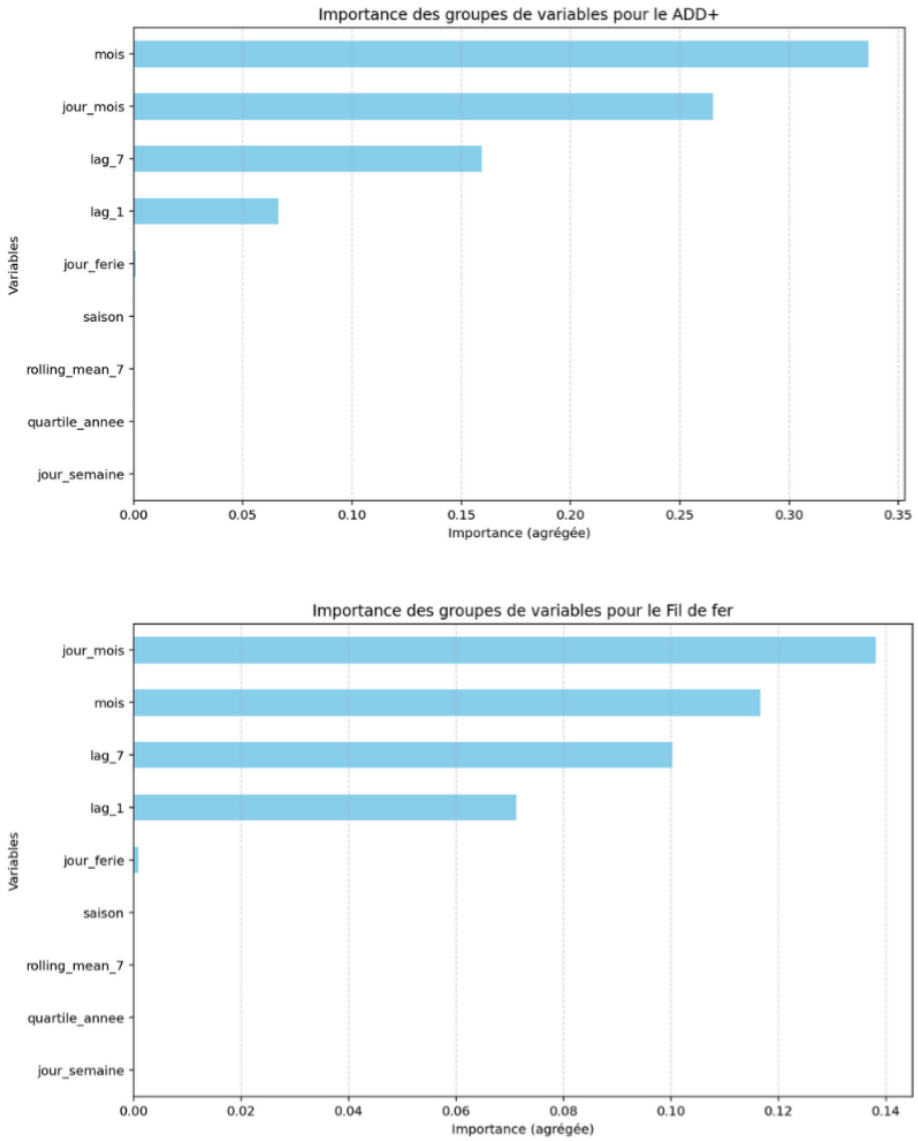

FIGURE 3.5 – Graphe comparatif entre L'importance des differentes variable sur Les prédiction de tout les produits en utilisant le modèle RandomForestRegressor (partie 1).

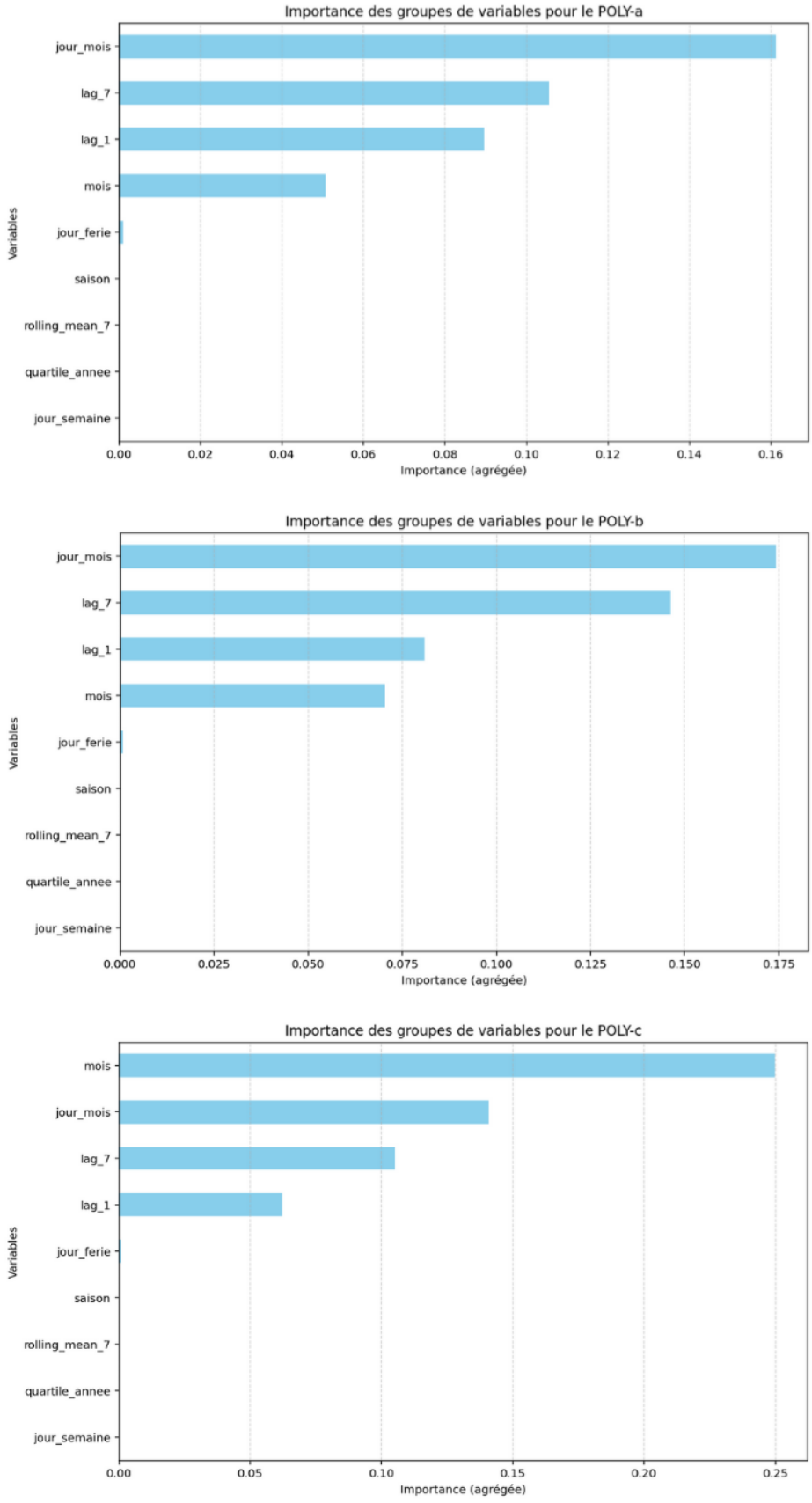

FIGURE 3.6 – Graphe comparatif entre L'importance des differentes variable sur Les prédiction de tout les produits en utilisant le modèle RandomForestRegressor (partie 2).

3.3.2.1 Analyse de l'Importance des Caractéristiques

Le Tableau 3.2 présente l'importance des différentes variables utilisées par le modèle Random Forest Regressor pour prédire la consommation de chaque article. Les importances sont normalisées de sorte à totaliser 1 pour chaque modèle.

Caractéristiques Importantes Communes : Les variables *mois* et *jour_mois* ressortent comme systématiquement importantes, traduisant l'existence de tendances mensuelles et quotidiennes marquées dans les données de consommation. Par exemple, *mois* est particulièrement influent pour POLY-c (0,2498) et ADD+ (0,3362), tandis que *jour_mois* l'est pour POLY-a (0,1612), POLY-b (0,1742) et ADD+ (0,2652).

Les variables *lag_1* (consommation de la veille) et *lag_7* (consommation de la semaine précédente) sont également importantes pour tous les articles. *lag_7* se distingue comme plus influent que *lag_1* dans la majorité des cas, ce qui indique des effets hebdomadaires significatifs, notamment pour POLY-b (0,1463) et ADD+ (0,1593).

Caractéristiques avec Importance Nulle ou Faible : Les variables *jour_semaine*, *saison*, *quartile_annee* et *rolling_mean_7* ont une importance nulle dans tous les cas. Cela suggère que leurs effets sont soit redondants (par exemple, la variable *mois* pouvant déjà capturer la saisonnalité), soit insuffisamment informatifs dans le cadre du modèle utilisé. L'absence d'importance pour *jour_semaine* peut être due à l'encodage ou au fait que *lag_7* intègre déjà les cycles hebdomadaires.

De même, la variable *jour_ferie* présente une importance très faible mais non nulle (entre 0,0007 et 0,0011), indiquant un effet marginal des jours fériés sur la consommation.

3.3.3 Long Short-Term Memory (LSTM)

Le modèle Long Short-Term Memory (LSTM) est un type de réseau de neurones récurrents (RNN) particulièrement adapté aux séries temporelles, c'est donc un choix assez évident(se référer au 1^{er} chapitre pour plus de détails).

3.3.3.1 Préparation des données

Tout comme pour l'algorithme RandomForest,le jeu de données est filtré pour ne conserver que l'article cible et un encodage cyclique a été appliqué aux variables temporelles.

Une normalisation MinMax a été appliquée afin de faciliter l'apprentissage du réseau. Les séquences d'entrée ont été construites en utilisant une fenêtre glissante de 30 jours. Autrement dit, chaque entrée du modèle est constituée des 30 jours précédents.

3.3.3.2 Architecture et entraînement

Afin d'optimiser les performances du modèle LSTM, une recherche d'hyperparamètres a été effectuée à l'aide de la bibliothèque Keras Tuner. Cette méthode a permis d'automatiser l'exploration de différentes combinaisons de paramètres tels que :

- Le nombre d'unités dans les couches LSTM (`lstm_units1`, `lstm_units2`),
- Le taux de régularisation par `Dropout`,
- Le choix de l'optimiseur (ici `adam`).

Le tuner a testé 10 configurations différentes (`max_trials=10`) sur un jeu d'entraînement avec une validation croisée interne (validation _split = 10%). Le meilleur modèle a été sélectionné selon la valeur minimale de la fonction de coût (MSE) sur les données de validation.

Le modèle implémente deux couches LSTM consécutives :

- une première couche avec 32 neurones, suivie d'un `Dropout` à 10%,
- une seconde couche LSTM à 32 neurones,
- une couche de sortie `Dense(1)` pour prédire la quantité future.

L'entraînement du modèle est réalisé à l'aide de l'optimiseur `Adam`, avec la fonction de coût erreur quadratique moyenne (MSE).
Une stratégie d'arrêt anticipé (*EarlyStopping*) est mise en place avec une patience de 50 itérations, afin de prévenir le sur-apprentissage.
Une validation croisée interne est réalisée sur 20% des données d'entraînement via l'option validation_split=0.2.

3.3.3.3 Résultats et évaluation

Le modèle a été évalué sur les données de l'année 2025, et ce sur cinq articles distincts (`POLY-a`, `POLY-b`,`POLY-c`,`ADD+`,`fil de fer`). Les métriques obtenues sont :

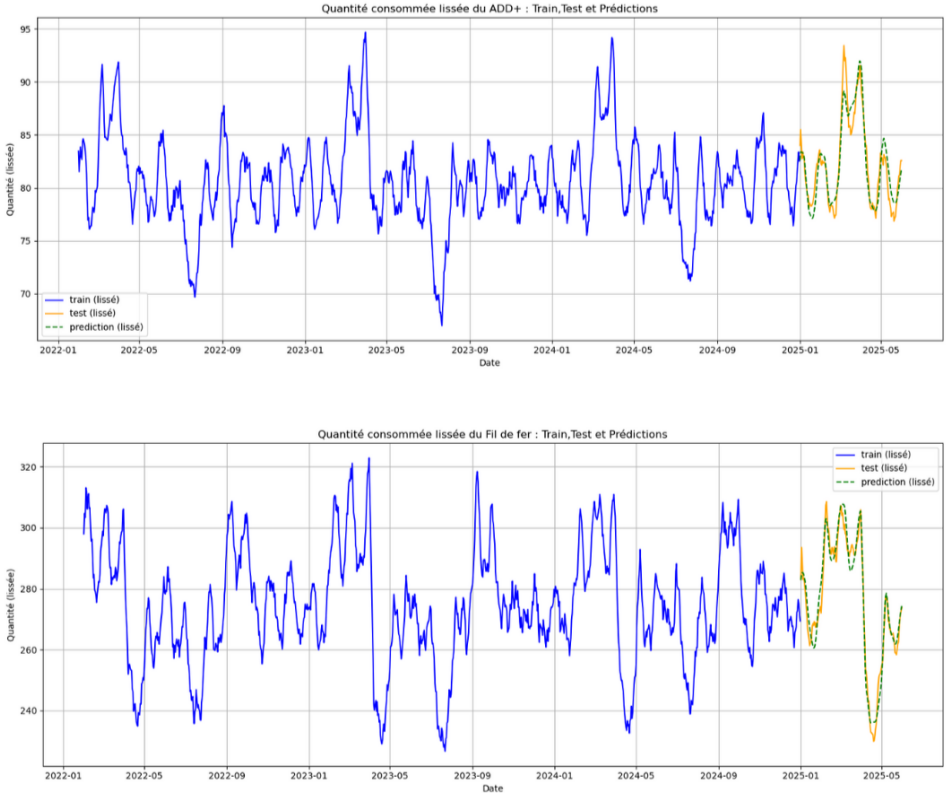

FIGURE 3.7 – Graphe comparatif entre les données réel et prédites en utiliser un modèle LSTM Pour chaque produits.(partie 1).

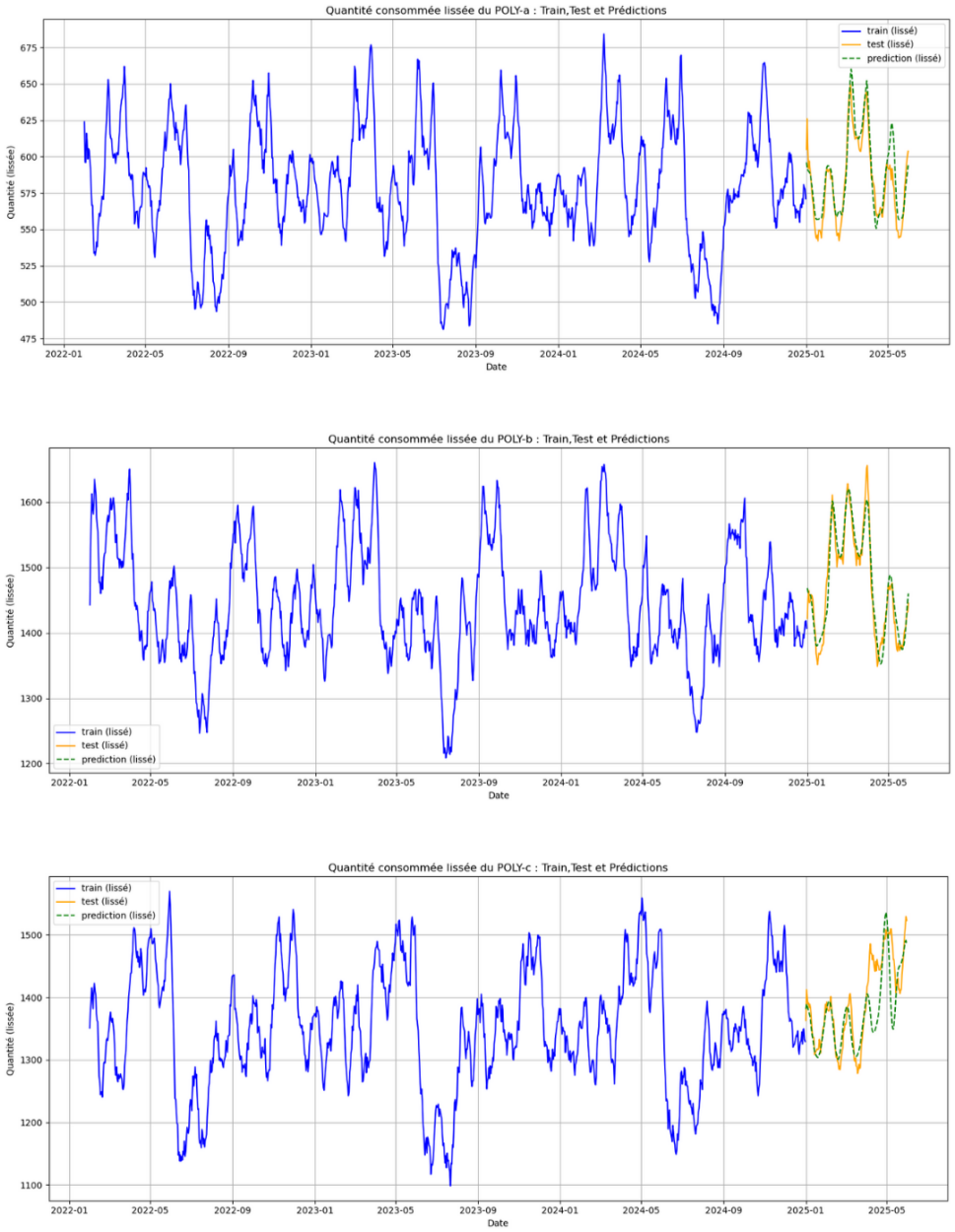

FIGURE 3.8 – Graphe comparatif entre les données réel et prédites en utiliser un modèle LSTM Pour chaque produits. (partie 2).

Article	RMSE	R^2
POLY-a	28.92	0.474
POLY-b	62.09	0.645
POLY-c	69.55	0.449
ADD+	3.80	0.542
fil de fer	12.49	0.742

TABLE 3.3 – Performance du modèle LSTM sur les articles testés (année 2025)

3.3.3.4 Analyse des Performances du Modèle

Les graphes 3.7 et 3.8 montrent les performances de prédiction des différents modèles LSTM (POLY-a, POLY-b, POLY-c, ADD+, Fil de fer) sur des séries chronologiques de "Quantité consommée lissée".

Tendances Générales : Pour tous les modèles, la courbe bleue représente les données d'entraînement (train), la courbe orange les données de test (test), et la courbe verte en pointillé les prédictions (prediction). On observe que les modèles parviennent généralement à capturer les tendances et les variations saisonnières des données d'entraînement.

Performance sur la Période de Test :

- POLY-a, POLY-b, POLY-c : Sur la période de test, les prédictions suivent la tendance générale des données de test , mais il y a des décalages et des erreurs notables, surtout au niveau des pics et des creux. Cela est particulièrement visible pour POLY-b et POLY-c où les prédictions semblent moins bien épouser les variations que pour POLY-a.
- ADD+ et Fil de fer : Ces deux produits semblent avoir une meilleure capacité à suivre les variations fines des données de test. Les courbes de prédiction (vert pointillé) sont plus proches des données réelles de test (orange), suggérant une meilleure adéquation.

Analyse des Métriques d'Évaluation (RMSE et R²) Le tableau 3.3 fournit les métriques d'évaluation (RMSE et R²) du modèle pour chaque produit.

1. **En terme de RMSE :**

- DD+ (3.80) est le meilleur en termes de RMSE, suivi de "Fil de fer" (12.49).
- POLY-b (62.09) et POLY-c (69.55) ont les RMSE les plus élevés, indiquant les plus grandes erreurs de prédiction. Cela confirme visuellement ce que nous avons observé sur les graphes pour ces modèles.
- POLY-a (28.92) a un RMSE significativement plus élevé que ADD+ et Fil de fer, mais est meilleur que POLY-b et POLY-c.

2. **En terme de R^2 :**
   - Fil de fer (0.742) est le meilleur en termes de R^2, ce qui signifie que le modèle explique la plus grande partie de la variance des données.
   - POLY-b (0.645) et ADD+ (0.542) suivent, indiquant des capacités explicatives modérées.
   - POLY-a (0.474) et POLY-c (0.449) ont les R^2 les plus faibles, suggérant que le modèle explique moins bien la variabilité des données.

3.3.4 SHAP

Afin de mieux comprendre l'influence des variables d'entrée sur les prédictions du modèle LSTM, nous avons utilisé la méthode SHAP (SHapley Additive exPlanations). SHAP est une technique d'explicabilité basée sur la théorie des jeux de Shapley, qui permet d'attribuer une importance à chaque variable en fonction de sa contribution individuelle à la prédiction.

Nous avons utilisé `KernelExplainer`(Le KernelExplainer utilise la méthode KernelSHAP pour calculer les valeurs SHAP, vu un peu plus en détails dans le 2^{eme} chapitre) de la librairie SHAP, qui est une méthode modèle-agnostique, adaptée ici pour estimer les contributions des variables d'entrée du modèle LSTM. Pour cela :

- Un sous-échantillon de 100 observations du jeu d'entraînement a été utilisé comme fond de référence (`X_background`).
- 20 observations du jeu de test ont été utilisées pour le calcul des valeurs SHAP (`X_sample`).

Les valeurs SHAP ont ensuite été moyennées pour chaque variable afin de produire un graphique d'importance globale.

feature	POLY-a	POLY-b	POLY-c	ADD+	fil de fer
jour_semaine	0.000000	0.000000	0.000000	0.000000	0.000000
saison	0.000000	0.000000	0.000000	0.000000	0.000000
mois	0.000085	0.000883	0.000016	0.000413	0.000065
jour_ferie	0.000644	0.001071	0.000296	0.000402	0.000455
quantite	0.000830	0.002592	0.000234	0.000352	0.001854
jour_mois	0.001941	0.000685	0.001424	0.002861	0.002073

TABLE 3.4 – Importance des variables du modèle représenté avec la moyenne des valeur de SHAP (année 2025)

FIGURE 3.9 – Graphe comparatif entre Les influence des differente variable sur Les prediction sur tout les produits.

3.3.4.1 Analyse de l'Importance des Features

Le tableau 3.4 et les graphes 3.9 montrent l'importance moyenne absolue de chaque feature et de chaque produit.

Observations Générales :

jour_semaine" et "saison" ont systématiquement une importance nulle, pour tous les produits (0.000000). Cela suggère que ces caractéristiques n'ont aucun impact sur les prédictions des modèles LSTM dans ce contexte.

Cela pourrait indiquer qu'elles ne sont pas des prédicteurs significatifs de la "quantité consommée" ou que leur information est déjà capturée par d'autres features.
"mois" a également une très faible importance pour la plupart des produits.
"jour_mois" et "quantite" sont les features les plus influentes globalement sur la performance du modèle. Leur importance varie entre les produits, mais elles se classent constamment parmi les meilleures.
"jour_ferie" est également une feature modérément importante.

Détail par produit :

- POLY-a :"jour_mois" est la feature la plus importante (0.001941), suivie de "quantite" et "jour_ferie".
- POLY-b :"quantite" est de loin la feature la plus importante (0.002592).
- POLY-c :"jour_mois" est la feature la plus importante (0.001424), suivie de "jour_ferie" et "quantite".
- ADD+ :"jour_mois" est la feature la plus importante (0.002861), suivie de "quantite" et "jour_ferie".
- Fil de fer :"jour_mois" est la feature la plus importante (0.002073), suivie de "quantite" et "jour_ferie".

Conclusion

Ce chapitre a permis de mettre en œuvre concrètement les méthodes de science des données dans un contexte industriel réel, à travers le cas d'étude de l'entreprise BATELEC. La problématique de la prévision de la consommation journalière de matières premières a été abordée de manière systématique, depuis la préparation des données jusqu'à l'interprétation des résultats.

Deux approches complémentaires ont été explorées : un modèle basé sur les forêts aléatoires (Random Forest), performant et facilement interprétable, et un modèle de réseau de neurones récurrent (LSTM), dit plus adapté à la capture des dépendances temporelles complexes. L'interprétation des résultats via la méthode SHAP a permis de mettre en évidence les variables les plus influentes

dans les décisions des modèles, renforçant ainsi la transparence et la confiance dans les prédictions.

Conclusion générale

Ce mémoire a permis de mettre en lumière les enjeux actuels liés à l'interprétation des modèles de machine learning, dans un contexte où ces outils sont de plus en plus utilisés pour soutenir la prise de décision. Nous avons montré que la performance seule ne suffit plus, les modèles doivent désormais être compréhensibles, justifiables et transparents.

Après avoir posé les bases théoriques du machine learning et détaillé les principaux types d'algorithmes (supervisés, non supervisés, séquentiels), nous avons exploré différentes approches d'interprétation, qu'elles soient intrinsèques ou post-hoc. Les méthodes SHAP et LIME se sont révélées particulièrement puissantes pour comprendre le comportement de modèles comme les forêts aléatoires ou les réseaux LSTM.

L'étude de cas sur l'entreprise BATELEC a illustré concrètement ces méthodes, en montrant comment des techniques avancées de modélisation peuvent être couplées à des outils d'explication pour résoudre un problème industriel réel : la prévision de la consommation de matières premières. Les résultats obtenus montrent qu'il est possible de concilier performance et transparence.

Ce travail ouvre plusieurs perspectives : l'intégration de ces méthodes dans des outils de visualisation destinés aux décideurs, l'extension à d'autres types

de modèles, ou encore la prise en compte explicite des contraintes éthiques et réglementaires dans les systèmes intelligents. Plus globalement, ce mémoire illustre une tendance forte de l'IA moderne : rendre les modèles non seulement efficaces, mais aussi responsables et compréhensibles.

Bibliographie

[1] Serge Abiteboul, Peter Buneman, and Dan Suciu. *Data on the web : from relations to semistructured data and XML*. Morgan Kaufmann, 2000.

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[3] A. Aman. Using shap values to explain how your machine learning model works, 2024. Accessed : 2025-06-24.

[4] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2) :157–166, 1994.

[5] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis : forecasting and control*. John Wiley & Sons, 2015.

[6] Nadia Burkart and Marco F Huber. A survey on the explainability of supervised machine learning. *Journal of Artificial Intelligence Research*, 70 :245–317, 2021.

[7] Chris Chatfield. *The Analysis of Time Series : An Introduction*. CRC Press, 6 edition, 2004.

[8] Thomas H Davenport and Laurence Prusak. *Working knowledge : How organizations manage what they know*. Harvard Business Press, 1998.

[9] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv :1702.08608*, 2017.

[10] Mengnan Du, Ninghao Liu, and Xia Hu. Techniques for interpretable machine learning. *Communications of the ACM*, 63(1) :68–77, 2019.

[11] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful : Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, 20(177) :1–81, 2019.

[12] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow : Concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2022.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[14] Bryce Goodman and Seth Flaxman. Eu regulations on algorithmic decision-making and a "right to explanation". In *ICML workshop on human interpretability in machine learning (WHI 2016), New York, NY. http ://arxiv. org/abs/1606.08813 v1*, 2016.

[15] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning : data mining, inference and prediction. *The Mathematical Intelligencer*, 27(2), 2005.

[16] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786) :504–507, 2006.

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8) :1735–1780, 1997.

[18] Rob J Hyndman and George Athanasopoulos. *Forecasting : principles and practice*. OTexts, 2018.

[19] William H Inmon. *Building the data warehouse*. John wiley & sons, 2005.

[20] Zachary C Lipton. The mythos of model interpretability : In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3) :31–57, 2018.

[21] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[22] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1 : Statistics*, volume 5, pages 281–298. University of California press, 1967.

[23] Tim Miller. Explanation in artificial intelligence : Insights from the social sciences. *Artificial intelligence*, 267 :1–38, 2019.

[24] Tom M Mitchell and Tom M Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[25] Christoph Molnar. Interpretable machine learning. 2019, 2020.

[26] Max D Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2) :161–174, 1991.

[27] W James Murdoch, Chandan Singh, Karl Kumbier, Reza Abbasi-Asl, and Bin Yu. Interpretable machine learning : definitions, methods, and applications. *arXiv preprint arXiv :1901.04592*, 2019.

[28] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, 140 :110190, 2020.

[29] Foster Provost and Tom Fawcett. Data science and its relationship to big data and data-driven decision making. *Big data*, 1(1) :51–59, 2013.

[30] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you ?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[31] Stuart J Russell and Peter Norvig. *Artificial intelligence : a modern approach*. pearson, 2016.

[32] Zahra Sadeghi and Stan Matwin. A review of global sensitivity analysis methods and a comparative case study on digit classification. *arXiv preprint arXiv :2406.16975*, 2024.

[33] Andrea Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global sensitivity analysis : the primer*. John Wiley & Sons, 2008.

[34] Wojciech Samek, Grégoire Montavon, Sebastian Lapuschkin, Christopher J Anders, Klaus-Robert Müller, et al. Toward interpretable machine learning : Transparent deep neural networks and beyond. *arXiv preprint arXiv :2003.07631*, 2, 2020.

[35] Monica Scannapieco. *Data quality : concepts, methodologies and techniques. Data-centric systems and applications*. Springer, 2006.

[36] 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*, pages 618–626, 2017.

[37] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-cam : Why did you say that ? *arXiv preprint arXiv :1611.07450*, 2016.

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

[39] Matt Turek. Explainable artificial intelligence (xai). *Defense Advanced Research Projects Agency*, 2018.

[40] Richard Y Wang and Diane M Strong. Beyond accuracy : What data quality means to data consumers. *Journal of management information systems*, 12(4) :5–33, 1996.

Résumé

Dans un contexte où les modèles de machine learning deviennent de plus en plus complexes, la question de leur interprétabilité est devenue centrale. Ce mémoire propose une exploration des approches d'explication des modèles d'apprentissage automatique. Après une présentation des fondements théoriques de la science des données et des typologies de l'apprentissage automatique, nous mettons en œuvre des modèles de prédiction appliqués à un cas industriel réel. Des algorithmes comme Random Forest et LSTM ont été utilisés pour modéliser les données, tandis que les techniques d'interprétation ont permis de comprendre l'impact des variables sur les prédictions. Les résultats montrent que l'intégration de méthodes d'explicabilité améliore significativement la compréhension des modèles, sans compromettre leur performance.

Mots-clés : apprentissage automatique, science des données, série temporelle, SHAP, interprétabilité, explicabilité, apprentissage profond, LSTM, forêts aléatoires, importance des variables.

Abstract

As machine learning models become increasingly complex, the question of their interpretability has become central. This thesis explores various approaches to explain machine learning models. After presenting the theoretical foundations of data science and machine learning typologies, we implement predictive models on a real-world industrial case. Algorithms such as Random Forest and LSTM are used to model the data, while interpretation techniques are employed to understand the impact of input variables on predictions. The results show that integrating explainability methods significantly enhances model understanding without compromising performance.

Keywords : machine learning, data science, time series, SHAP, interpretability, explainability, deep learning, LSTM, random forests, feature importance.