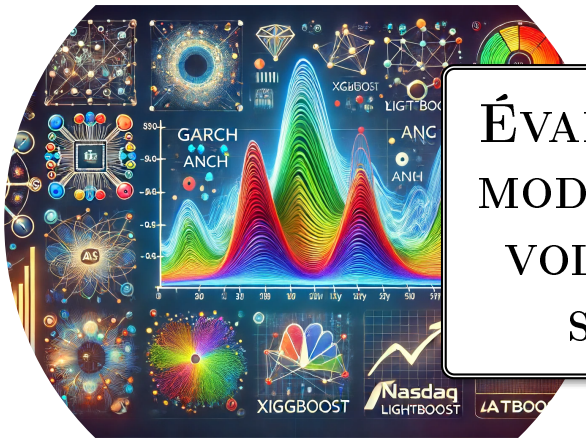
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة بجاية
Tasdawit n'Bgayet
Université de Béjaïa

Université Abderrahmane Mira de Béjaia
Faculté de Sciences Exactes
Département de Recherche Opérationnelle
Spécialité : Mathématiques Financières

Mémoire de master

THÈME



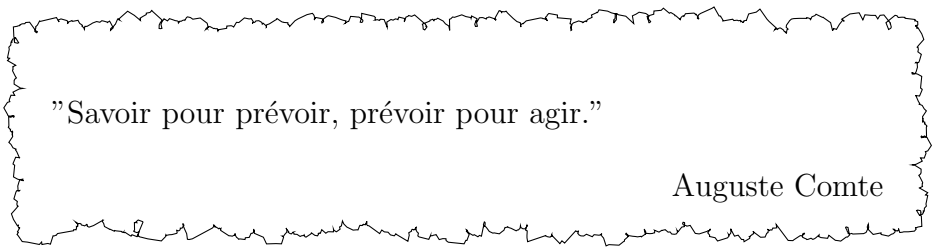
Évaluation comparative des modèles de prédiction de la volatilité dans différents secteurs économiques

Présenté par :
DJERROUD Nouara

Soutenu le 04/07/2024 devant le jury :

Président	Mme BOUIBED Karima	MCB
Encadreur	Mr TOUCHE Nassim	MCA
Examinatrice	Mlle AOUDIA Zohra	MAA
Examinateur	Mr BRAHMI Belkacem	MCA

Année universitaire:
2023/2024


"Savoir pour prévoir, prévoir pour agir."
Auguste Comte

Remerciements

Je souhaite exprimer ma gratitude à toutes les personnes qui ont contribué à la réalisation de ce mémoire.

Tout d'abord, je remercie mon encadrant Mr TOUCHE Nassim, pour ses conseils précieux, son soutien constant, et sa patience tout au long de ce projet. Ses encouragements et son expertise ont été essentiels pour mener à bien ce travail.

Je remercie également les membres du jury pour avoir pris le temps de lire et d'évaluer ce mémoire. Leurs observations et suggestions sont très appréciées.

Je tiens à remercier tout mes professeurs pour leur enseignement et leur soutien durant mes études. Leur dévouement et leur passion m'ont beaucoup inspiré.

Un grand merci à mes parents, pour leur amour, leur patience et leur soutien. Leur confiance en moi m'a donné la force de persévérer.

Je souhaite également remercier mon frère et ma sœur pour leur aide précieuse tout au long de ce projet. Leur soutien a été inestimable. Et un merci spécial à mon petit neveu, dont sa joie et son énergie m'ont donné le courage de continuer. Votre présence m'a apporté beaucoup de bonheur et de motivation.

Je tiens à exprimer une gratitude particulière à mon chat, qui m'a accompagné pendant mes nombreuses nuits de révision. Sa présence rassurante et affectueuse m'a apporté un soutien précieux et un réconfort constant.

Enfin, je remercie toutes les personnes qui, de près ou de loin, ont contribué à ce mémoire. Votre aide et votre soutien ont été très précieux.

Dédicace

Je dédie ce projet:

Je dédie ce mémoire à tous ceux qui ont contribué à mon parcours académique et personnel.

À maman et papa, pour leur soutien inconditionnel et leur amour indéfectible.

À ma sœur, mon frère, et mon neveu, pour leur soutien moral et leurs moments de détente nécessaires.

À mes professeurs, pour leurs conseils avisés et leur encouragement constant.

À mon chat fidèle, pour sa compagnie réconfortante durant les longues nuits de révision.

Enfin, je dédie ce travail à toutes les personnes qui croient en l'importance de la persévérance et de la détermination.

Nouara

TABLE DES MATIÈRES

LISTE DES ABRÉVIATIONS

ADF Augmented Dickey-Fuller.
AIC Akaike Information Criterion.
AMEX American Stock Exchange.
ANN Artificial Neural Network.

CART Classification and Regression Trees.
CatBoost Categorical Boosting.

EFB Exclusive Feature Bundling.
EGARCH Exponential Generalized Autoregressive Conditional Heteroskedasticity.

GARCH Generalized Autoregressive Conditional Heteroskedasticity.
GBM Gradient Boosting Machine.
GOSS Gradient-based One-Side Sampling.

JB Jarque-Bera.

LightGBM Light Gradient Boosting Machine.

MAE Mean Absolute Error.
MCMC Markov Chain Monte Carlo.
ML Machine Learning.
MLE Maximum Likelihood Estimation.
MSE Mean Squared Error.

NASDAQ National Association of Securities Dealers Automated Quotations.
NYSE New York Stock Exchange.

RMSE Root Mean Squared Error.

SIC Standard Industrial Classification.

TS Target Statistic.

XGBoost Extreme Gradient Boosting.

TABLE DES FIGURES

LISTE DES TABLEAUX

INTRODUCTION GÉNÉRALE

La volatilité des marchés financiers est un élément crucial pour les investisseurs et les gestionnaires de risques. Elle reflète l'incertitude et le risque associés aux fluctuations des prix des actifs, et sa prédiction est essentielle pour prendre des décisions éclairées en matière de gestion de portefeuille, de tarification des produits dérivés et de couverture des risques[20].

Traditionnellement, les modèles Generalized Autoregressive Conditional Heteroskedasticity (GARCH) (Generalized Autoregressive Conditional Heteroskedasticity) étaient largement utilisés pour prédire la volatilité. Ces modèles, bien adaptés pour capturer la volatilité et ses effets de levier, sont limités par leur difficulté à gérer des données complexes et non linéaires[29]. L'essor des technologies de l'apprentissage automatique a ouvert de nouvelles perspectives pour la prédiction de la volatilité. Les réseaux de neurones artificiels (ANN) et les modèles de boosting modernes (XGBoost, LightGBM et CatBoost) offrent une flexibilité accrue et une capacité à gérer des données volumineuses et complexes, ce qui les rend potentiellement plus performants que les modèles GARCH[1].

L'étude de Curtis Nybo [29] a déjà comparé les performances des modèles GARCH et ANN pour la prédiction de la volatilité dans différents secteurs économiques. Cette étude a mis en lumière les limites des modèles traditionnels face à la complexité des données financières. Cependant, la montée en puissance des modèles de boosting modernes soulève une nouvelle question : ces modèles, reconnus pour leur capacité à gérer des données complexes, surpassent-ils les modèles traditionnels en termes de précision et de robustesse pour la prédiction de la volatilité dans différents secteurs ?

Ce mémoire vise à comparer les performances de ces différents modèles de prédiction de la volatilité en se focalisant sur leur application dans divers secteurs économiques. En utilisant des données provenant de cinq secteurs industriels (biens de consommation durables, santé, technologie, industrie manufacturière et autres), nous évaluons la robustesse et l'efficacité de chaque modèle en utilisant des métriques d'évaluation telles que Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) et Mean Squared Error (MSE).

L'objectif de ce mémoire est d'identifier le modèle le plus précis pour chaque secteur et d'apporter des recommandations pour la prédiction de la volatilité. Ce faisant, nous contribuerons à une meilleure compréhension des dynamiques de volatilité des marchés financiers et à une meilleure prise de décision en matière de gestion de risques.

Ce mémoire est structuré comme suit :

1. Chapitre 1 présente les modèles traditionnels de prédiction de la volatilité, ainsi que les outils mathématiques fondamentaux.
2. Chapitre 2 détaille les modèles de prédiction en apprentissage automatique, en se focalisant sur les architectures de réseaux de neurones et les modèles de boosting.
3. Chapitre 3 offre une comparaison et une évaluation des performances des différents modèles, en s'appuyant sur des données empiriques.
4. Conclusion qui suggère les modèles les plus efficaces pour les futurs calculs de prévisions, afin d'aider à la prise de décision.

CHAPITRE 1

OUTILS MATHÉMATIQUES FONDAMENTAUX

1.1 Introduction

Dans ce chapitre, nous explorons les séries financières, puis nous examinons ces propriétés, comme la stationnarité et l'autocorrélation, et introduisons la volatilité, ses types et méthodes de prévision, y compris la différence entre les modèles statistiques (comme GARCH et EGARCH) et les modèles de machine learning.

1.2 Les séries chronologiques en finance

Définition 1.1. (Série temporelle)
Une série temporelle est une suite d'observations numériques (mesures) ordonnées dans le temps. Ces observations sont représentées par $\{X_{t_1}, X_{t_2}, \ldots, X_{t_n}\}$, où $(t_1, t_2, \ldots, t_n)$ sont les instants de mesure[43].

En finance, elles servent à décrire et analyser les phénomènes de marché, comme les prévisions et l'analyse des risques, ainsi qu'à calculer les prix des actifs, les taux de change et les options. Cette capacité d'analyser est essentielle pour les décisions des investisseurs et des gestionnaires de portefeuille, car elle leur permet d'optimiser leurs stratégies et de mieux évaluer les risques[43].

Définition 1.2. (Rendement)
Le rendement quotidien (hebdomadaire, mensuel ou annuelle) d'un actif financier est défini comme suit[10] :

$$R_t = \log \frac{P_t}{P_{t-1}}, \quad t > 1. \tag{1.1}$$

O*ù*, P_t est le prix d'un actif financier à l'instant t.

1.2.1 Propriétés des séries financières

Le rendement d'un actif représente les variations quotidiennes du logarithme du prix, ce qui vaut un gain (ou perte) relatif. Il s'avère ainsi plus utile pour l'investisseur que le prix lui-même, car il lui permet d'évaluer les bénéfices potentiels. C'est pourquoi les analyses financières se concentrent sur les rendements des actifs plutôt que sur leurs prix. Dans l'analyse des séries financières, Charpentier (2002) identifie les principales caractéristiques que nous examinerons successivement[43] :

1.2.1.1 Stationnarité

La stationnarité est essentielle dans l'analyse des séries chronologiques car elle simplifie la modélisation en supposant que les propriétés statistiques (l'espérance et la covariance) des données restent constantes dans le temps. Cela facilite la précision des prévisions et la prise de décision dans la finance.

1. **La stationnarité stricte**

   Définition 1.3. (Stationnarité stricte)
   Une série chronologique $(X_t)_{t\in T}$ est strictement stationnaire (fortement stationnaire), si pour tout $n \geq 1$ et tout $t_1 < t_2 < \ldots < t_n$, ces vecteurs ont la même loi[24] :

   $$(X_{t_1}, X_{t_2}, \ldots, X_{t_n}) \quad et \quad (X_{t_1+k}, X_{t_2+k}, \ldots, X_{t_n+k}).$$

   et ce pour tout décalage temporel $k \in \mathbb{Z}$.

   Cette définition montre que la loi d'un processus strictement stationnaire ne dépend pas du temps.

2. **La stationnarité au second ordre**
   En pratique, il est important de pouvoir évaluer la loi du processus, mais cela peut être très difficile. C'est pourquoi on utilise souvent une forme de stationnarité plus faible et moins stricte.

   Définition 1.4. (Stationnarité faible)
   Une série chronologique $(X_t)_{t\in T}$ est stationnaire au second ordre ou (faiblement stationnaire) ou (simplement stationnaire) si sa moyenne m(t) et sa covariance $\Gamma(t,s)$ ne changent pas lorsqu'on les déplace dans le temps[24].

   (a)
   $$\mathbb{E}[X_t] = m(t) = m. \tag{1.2}$$

   (b)
   $$\mathbb{C}ov(X_t, X_s) = \Gamma(t,s) = \Gamma(t+k, s+k). \tag{1.3}$$

   pour tout $t, s \in T$ et pour tout décalage temporel $k \in \mathbb{Z}$.

   Ce qui implique que la variance du processus $(X_t)_{t\in T}$ est constante.

   $$\mathbb{V}(X_t) = \Gamma(t,t) = \Gamma(0,0). \tag{1.4}$$

D'o*ù*, l'espérance et la variance d'un processus stationnaire au second ordre sont constantes, tandis que sa fonction d'autocovariance dépend du décalage temporel entre les deux observations, tel que $h = t - s$

$$\Gamma(t,s) = \Gamma(t-s,0) = \Gamma(h,0) = \gamma(h). \tag{1.5}$$

Donc,

$$\mathbb{C}ov(X_{t+h},X_t) = \ldots = \mathbb{C}ov(X_{1+h},X_1) = \mathbb{C}ov(X_h,X_0) = \gamma(h). \tag{1.6}$$

Par exemple la fonction d'autocovariance du bruit blanc est la suivante :

$$\gamma(h) = \begin{cases} \sigma^2, & si \quad h = 0, \\ 0 & si \quad |h| \neq 0. \end{cases} \tag{1.7}$$

1.2.1.2 Autocorrélation

L'autocorrélation est fréquente dans les séries financières, ce qui montre que les valeurs passées affectent les valeurs futures. Cela est essentiel pour la modélisation et l'analyse prédictive.

Définition 1.5. (Autocorrélation)
Soit $(X_t)_{t\in T}$ une série chronologique stationnaire. Sa fonction d'autocorrélation $\rho(h)$ est définie par[18] :

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)}, \quad \forall h \in \mathbb{Z}. \tag{1.8}$$

ce qui montre également que l'autocorrélation d'un processus stationnaire dépend uniquement du décalage temporel h entre deux observations.

Donc, la fonction d'autocorrélation du bruit blanc est décrite par :

$$\rho(h) = \begin{cases} 1, & si \quad h = 0, \\ 0 & si \quad |h| \neq 0. \end{cases} \tag{1.9}$$

Proposition 1.2.1. *Soit $(X_t)_{t\in T}$ une série chronologique stationnaire. Alors, sa fonction d'autocovariance et sa fonction d'autocorrélation sont symétriques[18].*

1.2.1.3 Hétéroscédasticité

Les séries financières présentent souvent une hétéroscédasticité, où la variance des erreurs varie dans le temps. Les modèles comme GARCH sont utilisés pour modéliser cette caractéristique[10].

1.2.1.4 Queues épaisses

En pratique, on observe que les distributions des rendements logarithmiques ont des queues plus larges que celles des distributions normales.
La décroissance d'une loi de distribution peut être décrite par son exposant de queue.

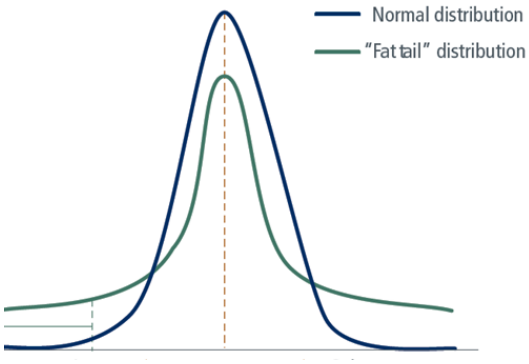


FIGURE 1.1 – Distributions des rendements logarithmiques et distributions normales.

Définition 1.6. (Exposant de queue)
On définit l'exposant de queue de la loi de distribution X par le nombre réel q_c, qui peut prendre la valeur infinie, tel que[24] :

$$q_c = \sup_q \left\{ \mathbb{E}[X^q] < +\infty \right\}. \tag{1.10}$$

On peut classifier les distributions en trois types[24] :

- Les distributions à queue légère, avec un exposant de queue $q_c = 1$, où tous les moments de la distribution existent et où la décroissance de la fonction de répartition est exponentielle dans les queues.
- Les distributions à queue lourde (ou épaisse), avec un exposant de queue $q_c < 1$, caractérisées par une décroissance de la fonction de répartition selon une certaine loi de puissance.
- Les distributions bornées, dont les valeurs sont incluses dans un ensemble borné, et par conséquent, ne possédant pas de queue.

Pour vérifier si les variations quotidiennes du rendement logarithmique ne suivent pas une distribution gaussienne, nous allons calculer la skewness et la kurtosis :

Définition 1.7. (Skewness)
La skewness est une mesure de l'asymétrie d'une distribution statistique[24] :

$$Skew(X) = \frac{\mathbb{E}[\mathbb{X} - \mathbb{E}[\mathbb{X}]]}{(\mathbb{E}[\mathbb{X} - \mathbb{E}[\mathbb{X}]])^{\frac{3}{2}}}. \tag{1.11}$$

Définition 1.8. (Kurtosis)
La kurtosis évalue la leptokurticité ou la non-gaussianité d'une distribution[24] :

$$Kurt(X) = \frac{\mathbb{E}[\mathbb{X} - \mathbb{E}[\mathbb{X}]]}{(\mathbb{E}[\mathbb{X} - \mathbb{E}[\mathbb{X}]])^{2}}. \tag{1.12}$$

Le coefficient d'aplatissement (kurtosis) de cette densité est supérieur à celui de la loi normale, qui est égale à trois. Le coefficient d'asymétrie (skewness) d'une loi normal est nul, tandis que la densité des rendements est généralement négatif. Cela signifie que la distribution est plus étroite et a des extrémités plus épaisses que celles d'une distribution normale.

1.2.1.5 Clustering de volatilité

Une caractéristique importante des séries financières est le "volatility clustering" ou accumulation de la volatilité. Cela signifie qu'il y a des périodes de forte volatilité et d'autres de faible volatilité. Ce phénomène est dû aux corrélations dans les séries financières, où un grand changement de valeur (comme une forte hausse ou baisse des prix) est souvent suivi par un autre grand changement, mais pas nécessairement dans la même direction (c'est-à-dire que le prix peut continuer à fortement augmenter ou diminuer). De même, lorsqu'un petit changement de valeur se produit, il est probable qu'il soit suivi par un autre petit changement. Ces périodes de forte ou faible volatilité peuvent durer des durées variables, car il n'y a pas de durée spécifique pour ces corrélations[27].

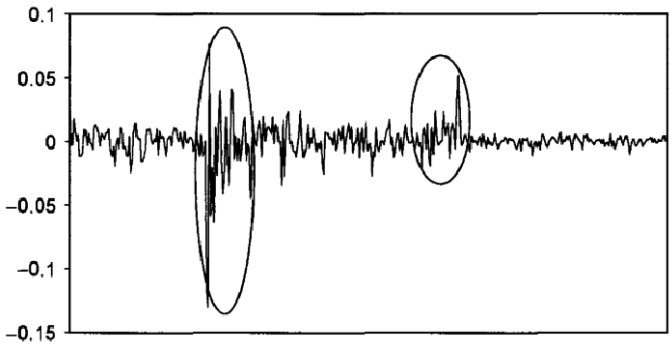


FIGURE 1.2 – Accumulation des volatilités.

1.2.1.6 Effet de levier

Les effets asymétriques dans la dynamique de la volatilité sont souvent appelés "effet de levier". Ce terme vient de Black, qui avait remarqué que les rendements des actifs financiers sont négativement corrélés avec les changements de leur volatilité. C'est à dire[27] :

— Quand les rendements sont plus bas (les prix baissent), la volatilité tend à augmenter.

— Quand les rendements sont plus hauts (les prix montent), la volatilité tend à diminuer.

Quelle que soit la raison de cette corrélation négative, elle introduit une asymétrie dans la distribution des rendements. Cette asymétrie qui est mesurée par "skewness".

La fonction de levier pour plusieurs actions :

$$\Gamma(t,s) = \frac{\mathbb{E}[\sigma_T X_t (\sigma_T X_s)^2]}{\mathbb{E}[(\sigma_T X_t)^2]}. \tag{1.13}$$

— Cette fonction est négative quand $t < s$: les rendements passés influencent la volatilité future, on voit une relation négative. Cette relation peut être décrite par une fonction exponentielle.
— Cette fonction est presque nulle quand $t > s$: la volatilité passée influence les rendements futurs, on ne trouve pas de relation claire. Cela signifie que les fluctuations des prix dans le passé n'ont pas vraiment d'impact sur les changements de prix futurs.

Les auteurs expliquent que cette asymétrie n'est pas due à des raisons économiques. Elle se produit parce que les investisseurs mettent du temps à réagir aux changements de prix.

1.3 La volatilité

La volatilité est une mesure statistique qui indique à quel point les rendements d'un titre de marché varient sur une certaine période. Elle se calcule généralement comme l'écart type des rendements. La volatilité est très importante en finance car elle aide à évaluer le risque des investissements, à fixer les prix des options et à gérer les portefeuilles d'investissement[20].

— **Haute volatilité :** Les prix changent beaucoup et rapidement. Opportunités de gains élevés, mais aussi de pertes importantes.
— **Basse volatilité :** Les prix changent peu et lentement. Donc y'a moins de risques, et plus de stabilité.

1.3.1 Types de volatilité en finance

1. **La volatilité historique** La volatilité historique se calcule en regardant les variations de prix passées d'un actif financier. Elle utilise les données historiques pour mesurer les fluctuations des rendements. Plus les prix ont beaucoup varié dans le passé, plus la volatilité historique est élevée.
2. **La volatilité implicite** La volatilité implicite se déduit des prix des options et reflète les attentes du marché sur les variations futures des prix. Elle est souvent vue comme un meilleur indicateur que la volatilité historique, car elle prend en compte les prévisions des investisseurs sur les événements futurs.
3. **La volatilité stochastique** Les modèles de volatilité stochastique supposent que la volatilité change de manière aléatoire au fil du temps. Ces modèles aident à comprendre les variations de la volatilité sur les marchés financiers et sont importants pour évaluer des produits financiers complexes.

1.4 La prévision de la volatilité en finance

La prévision de la volatilité permet de prédire les variations futures ou les changements des rendements d'un actif ou d'un portefeuille en utilisant des données passées. Pour ce faire, on utilise des modèles statistiques et des modèles d'apprentissage automatique pour faire ces prévisions. Ces estimations sont importantes car elles informent les investisseurs et les gestionnaires sur les futures variations de prix. Cela les aide à prévoir et à réduire les risques futurs

en présentant différents scénarios possibles et leurs probabilités. En conséquence, cela aide à prendre des décisions stratégiques.[27].

1.5 Modèles de prévision

1.5.1 Modèles statistiques

Les modèles statistiques sont des outils mathématiques fondamentaux utilisés pour analyser les données et prédire les résultats futurs ou estimer la probabilité d'événements. Ils sont largement utilisés dans divers domaines tels que la finance et le marketing. Ces modèles permettent de comprendre les relations entre différentes variables et de faire des prédictions basées sur des données passées, cette prédiction peut être réalisée de différentes manières selon le modèle choisi, telles que la régression, l'analyse de séries chronologiques et les modèles probabilistes comme GARCH. En générale, la prédiction se fait en se basant sur les tendances et les relations identifiées dans les données historiques, ce qui permet d'anticiper les valeurs futures ou les comportements à venir[25]. Cependant, les modèles statistiques fournissent une base solide pour la prise de décision en offrant des prédictions précises et en évaluant les risques potentiels.

1.5.1.1 Generalized autoRegressive conditional heteroskedasticity (GARCH)

Le modèle GARCH, qui signifie Hétéroscédasticité Conditionnelle Autorégressive Généralisée introduit par Bollerslev en 1986, est un modèle statistique utilisé en finance pour modéliser et prévoir la volatilité des séries chronologiques, en particulier les rendements financiers. Ce modèle est composé de deux parties : une composante autorégressive qui capture la volatilité passée et une composante de moyenne mobile qui s'ajuste aux chocs du système. Les modèles GARCH sont très utilisés en finance pour modéliser la volatilité des prix des actifs, gérer les risques et déterminer les prix des options. Un des avantages majeurs des modèles GARCH est leur flexibilité pour capturer les variations de la volatilité au fil du temps, ce qui est particulièrement utile pour identifier les périodes de forte et de faible volatilité dans les données financières[40].

Le modèle GARCH(p, q) modélise la variance conditionnelle suivante[26] :

$$X_t = \sigma_t \epsilon_t, \quad \epsilon_t \rightsquigarrow N(0,1). \tag{1.14}$$

O*ù*, ϵ_t est un bruit blanc.

$$\begin{aligned} \mathbb{V}[X_t|X_{t-1}] &= \sigma_t^2 && (1.15)\\ \sigma_t^2 &= \alpha_0 + \sum_{i=1}^{q} \alpha_i X_{t-i}^2 + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2. && (1.16)\\ &= \alpha_0 + A(L)X_t^2 + B(L)\sigma_t^2. && (1.17) \end{aligned}$$

Afin de garantir $\sigma_t^2 > 0$, les paramètres du modèle doivent respecter les contraintes suivante :

$$p \geq 0, \quad q > 0, \quad \alpha_0 > 0, \quad \alpha_i \geq 0 \quad avec \quad i = (1,\ldots,q), \quad \beta_i \geq 0 \quad avec \quad i = (1,\ldots,p).$$

Où L fait référence à l'opérateur de retard.

Le modèle GARCH est dit stationnaire si :

$$A(L) + B(L) < 1. \tag{1.18}$$

Dans ce cas,

$$\begin{aligned} \mathbb{E}[X_t] &= 0. \\ \mathbb{C}ov[X_t, X_s] &= 0, \quad pour \quad t \neq s. \end{aligned}$$

Ainsi, la variance inconditionnelle des modèles GARCH(p,q) est présentée comme suit :

$$\mathbb{V}[X_t] = \frac{\alpha_0}{1 - A(L) - B(L)}. \tag{1.19}$$

Soit un cas particulier du modèle GARCH(p,q), qui est le plus utilisé GARCH(1,1)[3] :

$$\sigma_t^2 = \alpha_0 + \alpha_1 X_{t-1}^2 + \beta_1 \sigma_{t-1}^2. \tag{1.20}$$

Avec,

$$\alpha_0 > 0, \quad \alpha_1 \geq 0, \quad \beta_1 \geq 0.$$

Lorsque la condition $\alpha + \beta < 1$ est satifaite, on obtient X_t stationnaire.
Donc, l'expression de la variance inconditionnelle est donné par :

$$\mathbb{V}[X_t] = \frac{\alpha_0}{1 - \alpha_1 - \beta_1}. \tag{1.21}$$

Estimation des paramètres GARCH
L'estimation des paramètres des modèles GARCH est importante en finance. La méthode la plus utilisée est le Maximum de Vraisemblance (Maximum Likelihood Estimation (MLE)), qui est précise mais demande beaucoup de calculs. Une autre méthode est la Méthode des Moments, qui est plus simple mais moins précise. L'estimation bayésienne est une autre approche puissante, utilise des algorithmes Markov Chain Monte Carlo (MCMC) (Markov Chain Monte Carlo) pour combiner les informations a priori et les données observées, et obtenir des distributions des paramètres. La Méthode des Moindres Carrés, moins fréquente, peut réduire les écarts entre les valeurs observées et prédites, bien qu'elle soit moins courante pour les modèles GARCH complexes. Le choix de la méthode dépend des données, des besoins d'analyse et des ressources disponibles[29].

Lorsqu'on doit ajuster un modèle hétéroscédastique aux données, on doit d'abord choisir l'ordre (p, q). Puis on estime les paramètres des modèles hétéroscédastiques avec la méthode MLE, en supposant toujours que les variables aléatoires ϵ_t sont indépendantes, identiquement distribuées, avec une moyenne nulle et une variance de un.

On pose les paramètres à estimer dans un vecteur $\theta = (\alpha_0, \alpha, \beta)^T$, avec la méthode de maximume vraisemblance, on déduit l'estimateur $\hat{\theta} = (\hat{\alpha_0}, \hat{\alpha}, \hat{\beta})^T$ [10].

$$L(\theta) = L(x_1, x_2, \ldots, x_n; \theta) = \prod_{t=1}^{n} f_\theta(\epsilon_t). \tag{1.22}$$

Avec,

$$f_\theta(\epsilon_t) = \frac{1}{\sqrt{2\pi\sigma_t^2}} \exp\Big(-\frac{x_t^2}{2\sigma_t^2}\Big). \tag{1.23}$$

$f_\theta(\epsilon_t)$ est la densité de ϵ_t.

Lorsque la fonction de vraisemblance L(.) est deux fois continuellement différentiable par rapport à θ pour tout $\theta \in \Theta$, alors l'estimateur du maximum de vraisemblance $\hat{\theta}$ est une solution du système suivant :

$$\Big(\frac{\partial L(\theta)}{\partial \theta}\Big)_{\theta=\hat{\theta}} = 0. \tag{1.24}$$

Et

$$\Big(\frac{\partial^2 L(\theta)}{\partial \theta^2}\Big)_{\theta=\hat{\theta}} < 0. \tag{1.25}$$

Dans le cas d'un modèle GARCH(1,1), le log vraisemblance se calcule à partir de :

$$L_T(\theta) = \sum_{t=1}^{T} l_t(\theta). \tag{1.26}$$

D'o*ù*,

$$l_t(\theta) = log\, f_\theta(\epsilon_t). \tag{1.27}$$

Alors,

$$l_t(\theta) = -\frac{1}{2}\log(2\pi) - \frac{1}{2}\log(\sigma_t^2) - \frac{1}{2}\Big(\frac{x_t}{\sigma_t}\Big)^2. \tag{1.28}$$

Ainsi, cette étape d'estimation nous permet de trouver les meilleurs paramètres pour le modèle hétéroscédastique, assurant qu'il s'adapte bien aux données.

Bien que les modèles GARCH fournissent des prévisions pour les périodes futures. En effet ces modèles sont symétriques. Pourtant, dans les séries financières, l'asymétrie est une hypothèse très réaliste. Ainsi, la famille des modèles GARCH a été enrichie par de nombreux modèles asymétriques dérivés des modèles GARCH, développés dans le but de résoudre ce problème d'asymétrie. Comme le modèle exponential GARCH (EGARCH) développé par Nelson. Ce modèle modélise le logarithme de la variance conditionnelle et permet également de prendre en compte l'effet de levier.

Le modèle EGARCH(p, q) est représenté par[38] :

$$log(\sigma_t^2) = \alpha_0 + \sum_{i=1}^{q} \alpha_i g(X_{t-i}) + \sum_{i=1}^{p} \beta_i log(\sigma_{t-i}^2). \tag{1.29}$$

O*ù*, X_t suit une loi normale est un bruit blanc et la fonction g vérifie :

$$g(X_{t-i}) = \theta X_{t-i} + \gamma(|X_{t-i}| - \mathbb{E}|X_{t-i}|). \tag{1.30}$$

Le coefficient $\gamma < 0$ représente l'effet de levier des chocs sur la volatilité.

La variance inconditionnelle EGARCH peut être calculée par :

$$\mathbb{V}[X_t] = \exp\left\{\frac{\alpha_0}{1-\sum_{i=1}^{p}\beta_i}\right\}. \tag{1.31}$$

L'avantage d'introduire un logarithme à la variance conditionnelle est qu'il n'est pas nécessaire d'imposer des contraintes sur les paramètres pour garantir que $\sigma_t^2 > 0$.

1.5.2 Modèles de machine learning

L'apprentissage automatique est un domaine de l'informatique qui crée des systèmes capables de s'améliorer par eux-mêmes grâce à l'expérience. Ces modèles deviennent plus efficaces en traitant de plus en plus de données. Ils sont surtout utilisés pour extraire des informations précieuses à partir de grandes quantités de données et pour améliorer les performances des machines. On utilise l'intelligence artificielle dans divers domaines tels que la santé, la finance , et le commerce. Par exemple, dans la finance, l'apprentissage automatique est utilisé pour détecter les fraudes, analyser les données de marché pour le trading algorithmique, gérer les risques, et optimisent les portefeuilles d'investissement[28].

Pour prédire des résultats, ces modèles analysent des données passées pour trouver des motifs et des relations, ce qui les aide à prendre des décisions avec de nouvelles données. L'entraînement des modèles implique de leur fournir des données historiques pour ajuster leurs paramètres et améliorer leurs capacités de prédiction[28].

1.5.2.1 Les étapes du traitement des données

Pour élaborer des algorithmes de Machine Learning (ML), il faut passer par plusieurs étapes[33] :

1. **Collecte de données :** Collectez des données importantes provenant de diverses sources (comme des fichiers de données, des flux de capteurs, des documents textuels, des images, des vidéos, des enregistrements audio, et des historiques d'achat) afin de les utiliser lors de la formation du modèle d'apprentissage automatique.
2. **Prétraitement des données :** Nettoyez les données en traitant les valeurs manquantes, en normalisant les caractéristiques et en transformant les variables catégorielles pour les préparer à l'entraînement du modèle.
3. **Sélection des caractéristiques :** Simplifier le modèle en ne conservant que des caractéristiques les plus importantes, ce qui peut améliorer sa précision et son efficacité tout en réduisant la complexité.
4. **Entraînement du modèle :** Utilisez les données prétraitées pour entraîner le modèle, afin qu'il puisse apprendre les motifs et relations présents dans les données.
5. **Évaluation du modèle :** Mesurez les performances du modèle en utilisant des critères d'évaluation pour vérifier sa précision et son efficacité dans les prévisions.

6. **Optimisation du modèle :** Ajustez les hyperparamètres du modèle pour maximiser ses performances et améliorer ses capacités de prédiction.

1.5.3 Comparaison des modèles de prévison

Tableau comparatif entre les modèles statistiques et les modèles d'apprentissage automatique[25].

Aspect	Modèles statistiques	Modèles d'apprentissage automatique
Approche de la modélisation	Basés sur des équations mathématiques et des hypothèses prédéfinies concernant la distribution des données.	Apprennent des modèles et des relations à partir des données.
Flexibilité	Moins flexibles, adaptés aux relations simples et linéaires dans les données.	Plus flexibles, peuvent gérer des relations complexes et non linéaires dans les données.
Interprétabilité	Plus interprétables, fournissent des informations sur l'influence de chaque variable sur le résultat.	Moins interprétables, peuvent être considérés comme des "boîtes noires" en raison de leur complexité et de leur manque de transparence dans les processus de prise de décision.
Taille des données	Peuvent avoir des difficultés avec de grands ensembles de données.	Peuvent gérer efficacement de grands ensembles de données et des données de grande dimension, adaptés aux applications de mégadonnées.

1.5.4 Les métriques de performance

Différentes mesures statistiques, telles que la moyenne des erreurs et l'erreur quadratique moyenne, sont utilisées pour évaluer les performances des modèles de prévision et déterminer leur efficacité dans la prédiction de la volatilité future. Elle mesurent la distance entre la valeur réelle et la valeur prédite[29] :

1. **L'erreur en moyenne absolue MAE (en anglais Mean Absolute Error)**

$$\mathrm{MAE} = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} |y_i - \hat{y}_i|. \tag{1.32}$$

2. **L'erreur en moyenne quadratique MSE (en anglais Mean Squared Error)**

$$\mathrm{MSE} = \frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} (y_i - \hat{y}_i)^2. \tag{1.33}$$

3. **L'erreur quadratique moyenne RMSE(en anglais Root Mean Squared Error)**

$$\text{RMSE} = \sqrt{\frac{1}{\text{N}} \sum_{i=1}^{\text{N}} (y_i - \hat{y}_i)^2}. \tag{1.34}$$

1.6 Conclusion

Ce chapitre a souligné l'importance de comprendre et de prédire la volatilité pour la gestion des risques financiers. Nous avons examiné différentes modèles de prévision et jugé leur efficacité, fournissant ainsi des métriques qui sont indispensables pour une prise de décision financière plus éclairée.

CHAPITRE 2

LES MODÈLES DE PRÉDICTION EN APPRENTISSAGE AUTOMATIQUE

2.1 Introduction

Dans ce chapitre, on présente différents modèles de prévision en machine learning, allant des réseaux de neurones artificiels (ANN), aux modèles avancées de boosting des arbres de décision Classification and Regression Trees (CART), comme XGBoost, LightGBM et CatBoost.

2.2 Réseaux de neurones artificiels

Les réseaux de neurones artificiels, ou ANN, sont des modèles informatiques inspirés par le fonctionnement du cerveau humain. Leur histoire remonte aux années 1940 et depuis, ils sont utilisés pour résoudre des problèmes de classification, de catégorisation, de prédiction, d'optimisation, et de reconnaissance. Ces modèles sont composés de nœuds interconnectés, similaires aux neurones du cerveau humain, et ils traitent les données en effectuant des opérations mathématiques complexes[8].

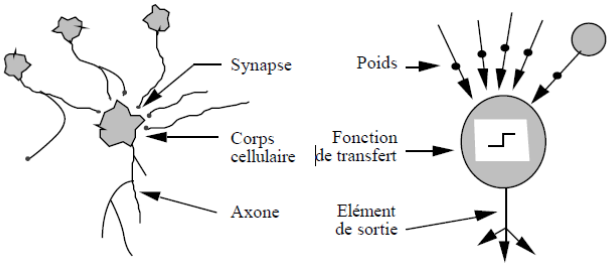


FIGURE 2.1 – Noeurone artificiel et noeurone biologique.

Pour comprendre les réseaux de neurones artificiels (ANN), il est essentiel de maîtriser quelques concepts fondamentaux. Les réseaux de neurones opèrent en deux phases : la conception et l'utilisation. Pendant la phase de conception, on détermine l'architecture du réseau, y compris le nombre de neurones et de couches cachées, ainsi que les fonctions d'activation pour chaque couche. Une fois cette architecture définie, le réseau est entraîné en ajustant les poids des connexions et les seuils de chaque neurone pour s'adapter à différentes conditions d'entrée. Ces ajustements permettent aux réseaux de neurones d'apprendre à partir des données et de prendre des décisions.[8].

Les applications des ANN sont nombreuses, allant de la reconnaissance d'images à la prévision financière. Leur capacité à résoudre des problèmes complexes en fait des outils précieux dans le domaine de l'intelligence artificielle et de la technologie moderne[8].

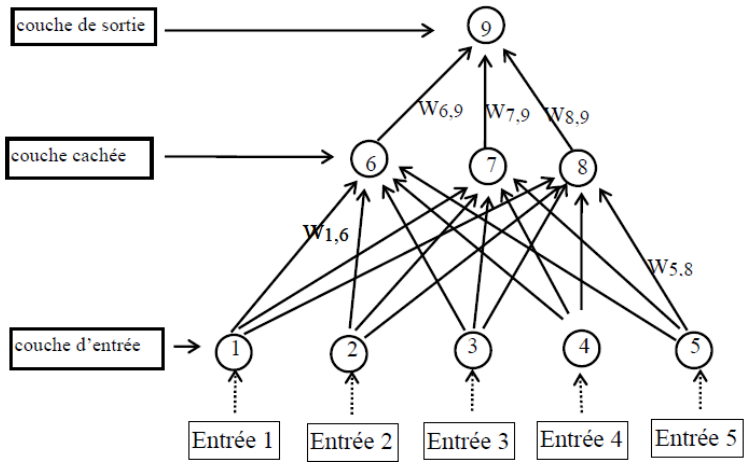


FIGURE 2.2 – Architecture d'un réseau à couches.

Dans ce schéma, le réseau comprend trois couches distinctes : une couche d'entrée qui reçoit les données à partir de cinq neurones, une couche de sortie avec un seul neurone fournissant le résultat final, et une couche intermédiaire non visible de l'extérieur, appelée couche cachée, qui effectue des calculs intermédiaires[31].
Le nombre de neurones dans la couche d'entrée dépend du nombre de variables d'entrée, tandis que le nombre de neurones dans la couche de sortie dépend du nombre de classes souhaitées. Le nombre de couches cachées et le nombre de neurones dans chaque couche cachée peuvent affecter les performances de généralisation du réseau. Un nombre trop petit de couches cachées ou de neurones peut entraîner une sous-performance du réseau, tandis qu'un nombre trop élevé peut entraîner un surajustement aux données[31].
Les connexions entre les neurones sont établies via des poids synaptiques, représentés par $w_{i,j}$. L'objectif de l'algorithme d'apprentissage est d'ajuster ces poids en fonction des données fournies pendant la phase d'apprentissage[31].

Il convient de mentionner que dans certains réseaux plus complexes, des connexions directes

peuvent être établies entre la couche d'entrée et la couche de sortie[31].

Les réseaux de neurones artificiels sont des modèles qui peuvent fonctionner à la fois en mode supervisé et non supervisé, qui sont définis comme suit[6] :

1. **Apprentissage supervisé :** L'apprentissage supervisé est une approche d'apprentissage automatique où un modèle est formé à partir d'exemples étiquetés. Dans ce processus, le modèle utilise un ensemble de données d'entraînement comprenant des entrées (caractéristiques) et des sorties correspondantes (étiquettes ou classes). L'objectif est d'apprendre une relation entre les caractéristiques et les étiquettes afin de prédire correctement les étiquettes pour de nouvelles données non étiquetées.
2. **Apprentissage non supervisé :** Lorsque seules des données non étiquetées sont disponibles et que les classes ainsi que leur nombre sont inconnus, on parle d'apprentissage non supervisé, également connu sous le nom de clustering. Dans ce contexte, l'apprentissage consiste à identifier des groupes homogènes d'exemples présents dans les données. Cela signifie trouver des groupes où les exemples les plus similaires sont regroupés ensemble et où les exemples les plus différents sont séparés dans des groupes distincts.

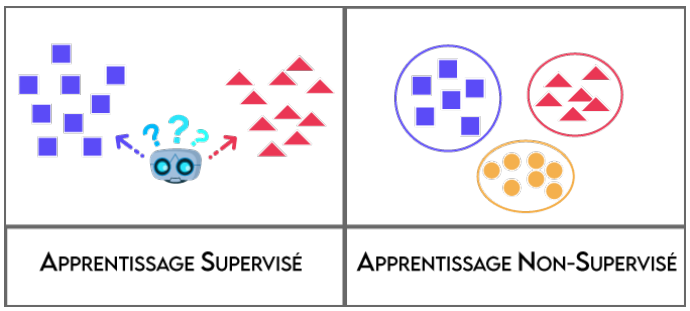


FIGURE 2.3 – Apprentissage supervisé et non supervisé.

2.2.1 Fonctionnement de l'algorithme des réseaux de neurones artificiels

1. **Initialisation des poids :** Les poids entre les neurones sont généralement initialisés de manière aléatoire pour commencer le processus d'apprentissage[32].
2. **Propagation avancée :** Les données d'entrée sont introduites dans la couche initiale, où chaque neurone reçoit des valeurs pondérées de la couche précédente et applique une fonction d'activation pour l'introduction de la non-linéarité[32].
3. **Calcul de la sortie :** La couche de sortie détermine la sortie finale en fonction des signaux propagés, la fonction d'activation variant selon le type de problème à résoudre[32].
4. **Calcul de l'erreur :** L'erreur du réseau est évaluée en comparant la sortie calculée à celle attendue, en utilisant différentes mesures d'erreur adaptées au type de problème[32].
5. **Rétropropagation :** L'algorithme de rétropropagation ajuste les poids des connexions en propageant les erreurs en arrière à travers le réseau, minimisant les erreurs à l'aide

de techniques d'optimisation telles que la descente de gradient[32].

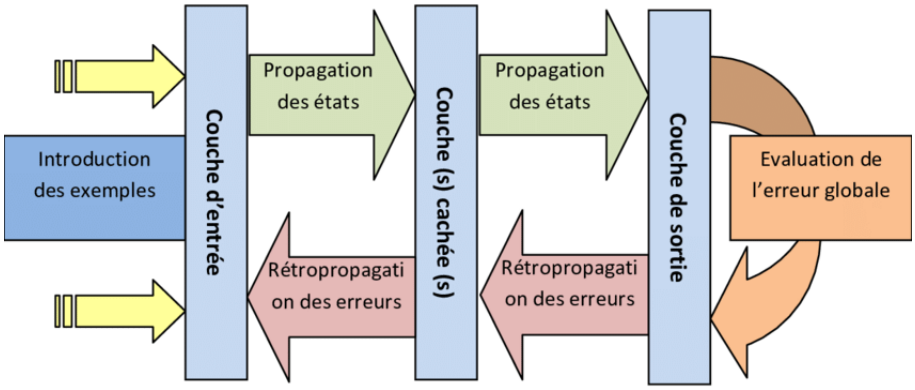


FIGURE 2.4 – L'algorithme de retropropagation.

6. **Répétition du processus :** Les étapes de propagation, de calcul des erreurs et de rétropropagation sont répétées sur plusieurs exemples d'apprentissage jusqu'à ce que le réseau converge pour minimiser les erreurs[32].
7. **Évaluation et ajustement :** Après l'entraînement, l'évaluation des performances du réseau sur des données de validation permet d'affiner les hyperparamètres pour améliorer son efficacité globale[32].

2.2.2 L'algorithme des réseaux de neurones artificiels

Définition 2.1. (neurone)
Un neurone est une fonction mathématique non linéaire, dont les paramètres sont limités à des valeurs bornées[8].

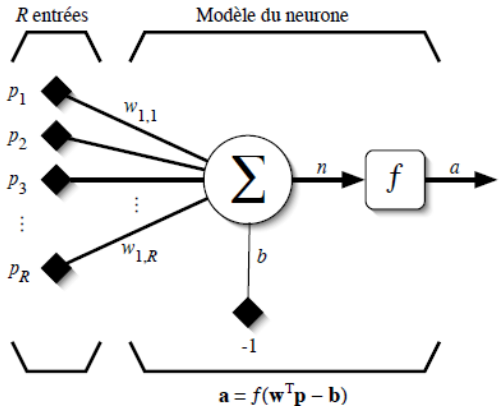


FIGURE 2.5 – Modèle d'un neurone artificiel.

2.2.2.1 La somme pondérée

Un neurone fonctionne en intégrant la somme pondérée de ses entrées. Cette somme, notée n, est ensuite transformée par une fonction d'activation f pour produire la sortie a du neurone. Les entrées du neurone sont représentées par un vecteur $p = [p_1, p_2, \ldots, p_R]^T$, tandis que les poids synaptiques transmettant l'activité du neurone j vers le neurone i, qui sont représentés par un vecteur $w = [w_{1,1}, w_{1,2}, \ldots, w_{1,R}]^T$.
L'expression de la sortie n de l'intégrateur est donnée par l'équation suivante[32] :

$$n = \sum_{j=1}^{R} w_{1,j} p_j - b. \tag{2.1}$$

Cette sortie n du neurone est obtenue en calculant une somme pondérée des entrées, à laquelle on soustrait le biais b du neurone. Cette somme pondérée est appelée le niveau d'activation du neurone. Le biais b est également connu sous le nom de seuil d'activation du neurone. Lorsque le niveau d'activation atteint ou dépasse le seuil b, l'argument de la fonction d'activation devient positif (ou nul). Sinon, il reste négatif.

2.2.2.2 Fonction de transfert

La somme pondérée est ensuite transformée par une fonction pour déterminer le potentiel de sortie.

Soit la fonction d'activation f pour obtenir la sortie a du neurone[32] :

$$a = f(n) = f(w^T p - b). \tag{2.2}$$

Les différentes fonctions de transfert peuvent être utilisées comme fonction d'activation du neurone, elles sont présentées dans le tableau suivant[32] :

Nom de la fonction	Relation d'entrée et sortie	Icône
seuil	$a = 0$ si $n < 0$ $a = 1$ si $n \geq 0$	
seuil symétrique	$a = -1$ si $n < 0$ $a = 1$ si $n \geq 0$	
linéaire	$a = n$	
linéaire saturée	$a = 0$ si $n < 0$ $a = n$ si $0 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire saturée symétrique	$a = -1$ si $n < -1$ $a = n$ si $-1 \leq n \leq 1$ $a = 1$ si $n > 1$	
linéaire positive	$a = 0$ si $n < 0$ $a = n$ si $n \geq 0$	
sigmoïde	$a = \frac{1}{1+\exp^{-n}}$	
tangente hyperbolique	$a = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	
compétitive	$a = 1$ si n est maximum $a = 0$ sinon	

On remarque, que la plupart des modèles actuels de réseaux de neurones sont des modèles à temps discret (synchrone), où le comportement des composants reste constant dans le temps.

En principe, toute fonction croissante et impaire peut être utilisée, mais il est courant d'utiliser des fonctions qui limitent la sortie à des bornes spécifiques. Parmi les choix possibles, la fonction sigmoïde, la fonction seuil, et linéaire, sont les plus fréquemment utilisées[31].

1. **La fonction seuil :** est une fonction qui applique un seuil sur son entrée. Si l'entrée est négative, la fonction retourne 0 (faux), sinon elle retourne 1 (vrai). Cette fonction est utilisée dans le contexte d'un neurone pour déterminer si la sortie doit être activée ou non. Le biais dans l'expression de la fonction détermine l'emplacement du seuil $w^{\mathrm{T}} p$ sur l'axe. Elle permet ainsi de prendre des décisions binaires.
2. **La fonction linéaire :** est une fonction très simple qui transmet directement son entrée à sa sortie. Dans ce cas, la sortie du neurone est simplement égale à son niveau d'activation, qui atteint zéro lorsque $w^{\mathrm{T}} p = b$.
3. **La fonction sigmoïde :** présente un comportement intermédiaire entre la fonction seuil et la fonction linéaire. Elle est non linéaire mais plus douce que la fonction seuil, qui est très abrupte. La sigmoïde ressemble à la fonction seuil lorsque les valeurs sont proches du seuil, et à la fonction linéaire lorsque les valeurs sont éloignées du seuil. Elle offre ainsi un compromis intéressant entre ces deux extrêmes.

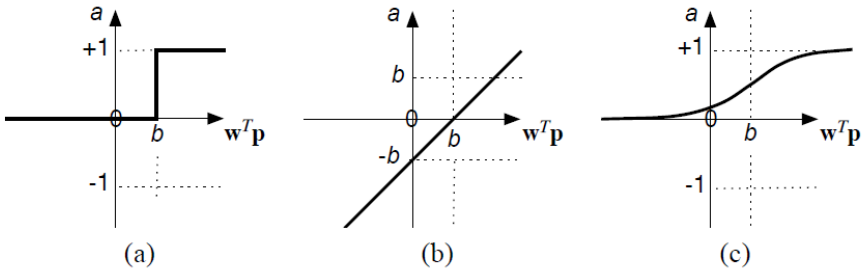


FIGURE 2.6 – Fonction de transfert : (a) du neurone «seuil» ; (b) du neurone «linéaire», et (c) du neurone «sigmoïde».

2.2.2.3 L'algorithme de rétropropagation

L'apprentissage supervisé par réseau de neurones consiste à ajuster les poids entre les neurones pour minimiser les écarts entre les valeurs de sortie attendues et celles prédites sur l'ensemble des données d'entraînement. Ce processus repose sur des exemples de données associés à des résultats attendus. L'algorithme de rétropropagation du gradient de l'erreur, développé par Rumelhart, Hinton et Williams en 1986[37], est largement utilisé, notamment dans le domaine financier, pour optimiser cet ajustement.

Pour un échantillon d'apprentissage contient s exemples, chacun décrit par un vecteur d'entrée $p_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,m}]^{\mathrm{T}}$, et un vecteur de sortie $\mathrm{D}_i = [d_{i,1}, d_{i,2}, \ldots, d_{i,n}]^{\mathrm{T}}$, la fonction que nous cherchons à minimiser est formulée comme suit[31] :

$$\mathrm{E} = \sum_{i=1}^{\mathrm{S}} \sum_{j=1}^{n} \frac{(a_{i,j} - d_{i,j})^2}{2}. \tag{2.3}$$

Les poids du réseau sont modifiés en suivant la formule suivante :

$$\Delta w_{i,j} = -\eta \frac{\partial \mathrm{E}}{\partial w_{i,j}}. \tag{2.4}$$

Le paramètre $0 < \eta < 1$, contrôle la vitesse de convergence de l'algorithme.

Après l'initialisation aléatoire des poids synaptiques, une première étape de calcul est réalisée dans l'ordre structurel du réseau. La sortie obtenue est ensuite comparée à la sortie souhaitée, évaluant ainsi la fonction de coût E. Ensuite, le système rétropropage l'erreur, $\frac{\partial \mathrm{E}}{\partial w_{i,j}}$ ce calcul se fait en partant de la couche de sortie vers la couche d'entrée, modifiant ainsi les poids synaptiques selon la formule précédemment énoncée. Ce processus se répète jusqu'à ce que l'utilisateur interrompe le processus ou qu'une valeur prédéfinie de la fonction de coût soit atteinte[31].

2.2.3 Principaux paramètres et mécanismes d'optimisation

Les hyperparamètres des réseaux de neurones artificiels[42] :

1. **Nombre de couches cachées (hidden_layers) :** Le choix du nombre de couches cachées impacte la capacité du réseau à apprendre des modèles et des relations complexes.
2. **Nombre de neurones dans chaque couche (neurons_per_layer) :** Le nombre de neurones dans chaque couche affecte la capacité du réseau à modéliser les données et ses capacités de généralisation.
3. **Taux d'apprentissage (learning_rate) :** Le taux d'apprentissage détermine la taille du pas lors de la mise à jour des poids et influence la vitesse et la stabilité de l'entraînement.
4. **Fonctions d'activation (activation_function) :** Le choix des fonctions d'activation telles que seuil, sigmoid ou linéaire impacte la capacité du réseau à capturer les non-linéarités dans les données.
5. **Techniques de régularisation (regularization)** L'utilisation de techniques de régularisation comme la régularisation L1/L2 aide à prévenir le surajustement et à améliorer la généralisation du modèle.
6. **Nombre maximum d'itérations (epochs) :** Détermine combien de fois l'algorithme traverse l'ensemble des données d'entraînement pendant l'apprentissage.
7. **Gradient minimum (min_gradient) :** Indiquer le moment où l'algorithme d'optimisation devrait s'arrêter, car un gradient inférieur à ce seuil signifie une convergence ou une stabilité suffisante dans la mise à jour des poids du modèle.

2.2.4 Les avantages et inconvénients

Les réseaux de neurones artificiels présentent plusieurs avantages et inconvénients[35] :

2.2.4.1 Avantages

Apprentissage automatique : Les réseaux de neurones artificiels permettent un apprentissage automatique, réduisant ainsi le besoin d'une intervention manuelle étendue dans la construction du modèle.

Résistance au bruit : Leur capacité à résister au bruit ou aux données peu fiables les rend appropriés pour manipuler des ensembles de données imparfaits.

Facilité d'utilisation : Les réseaux neuronaux sont plus simples à manipuler par rapport à l'analyse statistique traditionnelle, nécessitant moins d'efforts manuels.

Performance avec de petites données : Ils offrent de bonnes performances même avec des données limitées.

2.2.4.2 Inconvénients

Initialisation des poids : Le choix des valeurs de poids initiales et l'ajustement du taux d'apprentissage impactent significativement la vitesse de convergence.

Complexité : Les réseaux de neurones peuvent être complexes à optimiser en raison des nombreux paramètres impliqués, ce qui peut entraîner des temps d'entraînement plus longs et des exigences de calcul plus élevées.

L'absence de méthode systématique : la difficulté de déterminer de manière systématique la meilleure structure du réseau de neurones, comme le nombre de couches cachées et le nombre de neurones dans chaque couche cachée.

2.3 Arbres de décision

Les arbres de décision CART (Classification and Regression Trees) sont un modèle d'apprentissage automatique non paramétriques et non linéaires, largement utilisée dans des problèmes de classification et de régression. Ils constituent des modèles supervisés capables de prédire une variable cible discrète dite étiquetée (pour un problème de classification) ou continue (dans le cas d'une régression)[12]. Les arbres de décision, développés par Breiman en 1984 [5], sont souvent utilisés en combinaison avec d'autres techniques d'apprentissage automatique pour améliorer la précision des prédictions.

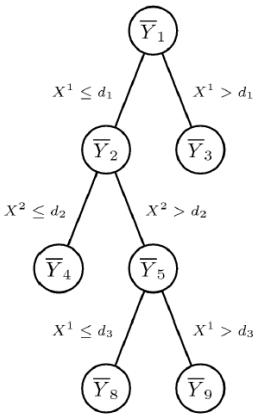


FIGURE 2.7 – Présentation d'un arbre dans le cas de la régression.

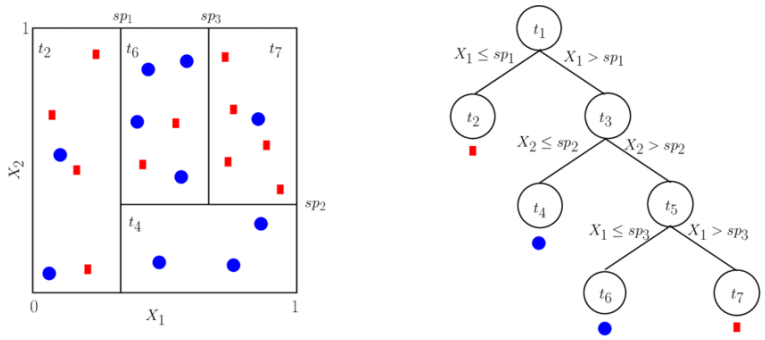


FIGURE 2.8 – Présentation d'un arbre dans le cas de la classification.

L'arbre de décision binaire est un modèle séquentielle où des décisions sont prises en divisant successivement la base de données en plusieurs groupes distincts[23]. Au début de l'approche, un arbre est caractérisé par un nœud principal, appelé la "racine de l'arbre", à partir de laquelle se trouve l'ensemble complet des données. Ensuite, chaque nœud est divisé en deux branches représentant une condition basée sur les caractéristiques des données, créant ainsi deux nœuds fils dont les sous-ensembles de données sont plus homogènes. Enfin, des feuilles marquent la fin du processus de division et représentent les résultats finaux. Cependant, la division s'arrête lorsqu'un critère d'arrêt est vérifié[7].

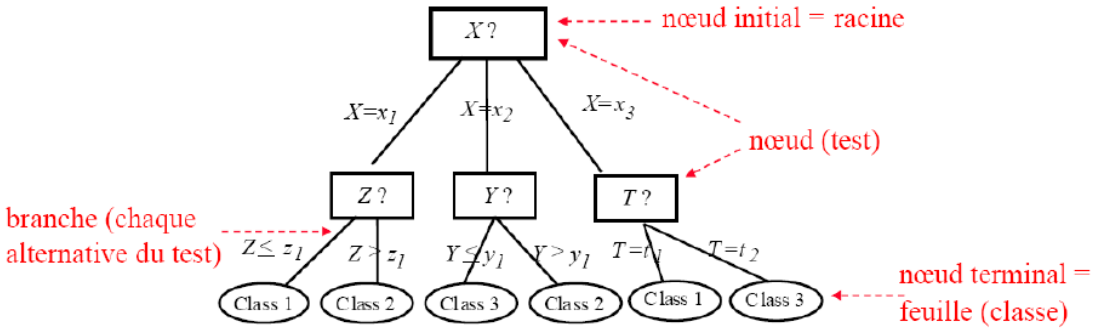


FIGURE 2.9 – Présentation d'un arbre décision.

2.3.1 La méthode d'agrégation d'arbres de décision

Différentes approches permettent de combiner plusieurs arbres de décision pour créer des modèles plus robustes. Nous nous concentrerons sur la méthode du Boosting, qui se distingue par son efficacité et sa capacité à améliorer les performances des modèles prédictifs. Développée dans les années 1990 par Robert Schapire et Yoav Freund[13], cette approche vise à surmonter les limitations des modèles individuels en agrégeant séquentiellement plusieurs modèles faibles pour former un modèle global plus puissant[39].

Le Boosting est une méthode d'apprentissage automatique largement utilisée dans divers domaines. Ce processus de Boosting commence par l'initialisation des poids sur les observations, généralement égaux pour toutes les observations. Ensuite, un premier modèle faible est construit et les poids des observations sont ajustés en fonction de ses performances. Les observations mal prédites par ce modèle initial reçoivent des poids plus élevés pour la construction du modèle suivant, tandis que les observations correctement prédites conservent leurs poids initiaux ou voient leur poids diminuer[23]. Ce processus se répète pour un certain nombre d'itérations, généralement défini à l'avance, jusqu'à ce qu'un classifieur fort soit obtenu en agrégeant les modèles faibles.

Le Boosting offre une grande flexibilité en termes de choix de modèle faibles et de fonctions de perte pour mesurer les performances, avec des modèles populaires comme XGBoost, LightGBM et CatBoost, qui se distinguent par leur approche de pondération pour renforcer l'apprentissage

des données mal ajustées et leur performance dans différentes tâches, telles que la classification et la régression[15].

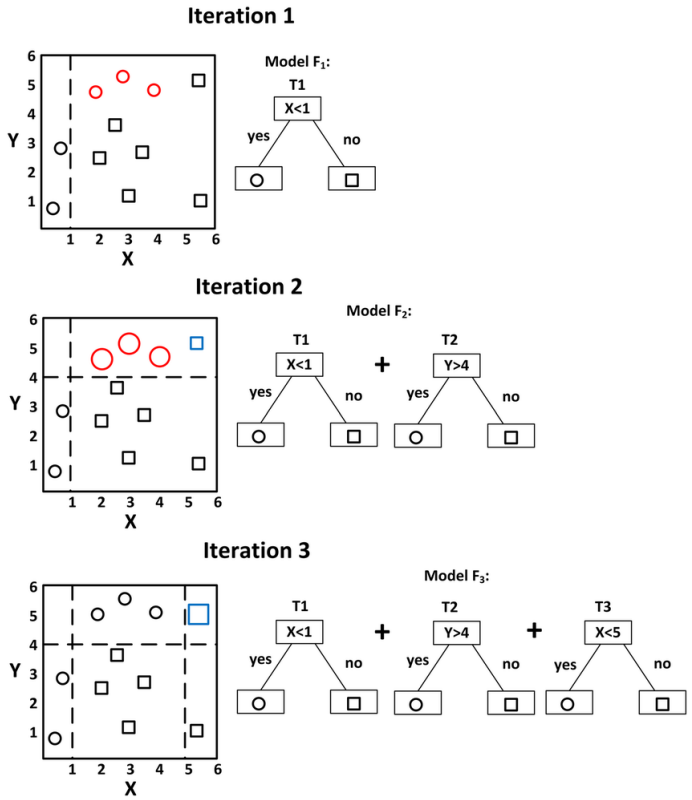


FIGURE 2.10 – Illustration de la méthode du boosting.

2.4 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (XGBoost), développé par Chen et Guestrin en 2016[9], est une implémentation de Gradient Boosting Machines (Gradient Boosting Machine (GBM)) proposée par Friedman en 2001[14], qui est modèle d'apprentissage supervisé. Ce modèle ensembliste, utilise le boosting pour agréger séquentiellement des estimateurs sur un échantillon d'apprentissage. Il ajuste les poids des individus en fonction de leurs performances, en donnant plus de poids aux observations mal prédites[4].

À chaque itération, un nouvel arbre apprend des erreurs du précédent, visant à minimiser à la fois le biais et la variance du modèle[23]. Pour cela, le modèle XGBoost utilise une méthode de descente de gradient pour optimiser une fonction de perte spécifiée, avec une régularisation intégrée pour prévenir le surajustement[4].

Ce modèle offre des fonctionnalités telles que la construction d'un modèle de machine learning robuste et précis, capable de s'adapter à une grande variété de problèmes de régression et de classification. Le modèle XGBoost est largement utilisé dans l'apprentissage automatique et l'analyse des données en raison de son efficacité, de sa flexibilité et de sa capacité à traiter des ensembles de données de grande taille[15].

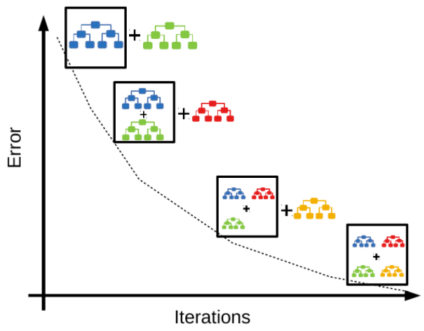


FIGURE 2.11 – Illustration du modèle ensembliste.

2.4.1 Fonctionnement de l'algorithme XGBoost

Le modèle XGBoost combine le boosting avec des arbres de décision. Son fonctionnement se déroule en deux étapes : tout d'abord, il construit une série d'arbres de décision faibles, puis il les combine pour former un modèle plus fort[33].

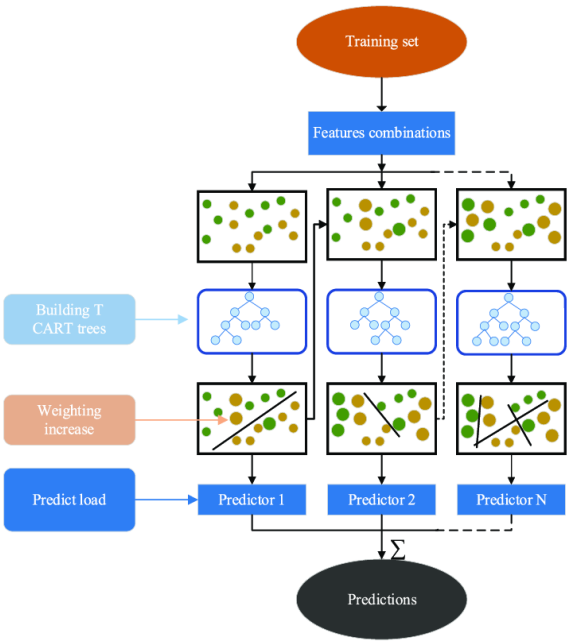


FIGURE 2.12 – Illustration du fonctionnement de l'algorithme de XGBoost.

1. **Construction des arbres de décision faibles :** Le modèle XGBoost utilise un ensemble d'arbres de décision faibles, construits séquentiellement en accordant un poids plus important aux observations mal prédites à chaque étape.
2. **Optimisation de la fonction de coût :** Pendant la construction des arbres, le modèle XGBoost optimise une fonction de coût en minimisant l'écart entre les prédictions du modèle et les vraies étiquettes des données. Cette fonction comprend un terme de perte pour mesurer l'erreur actuelle et un terme de régularisation pour éviter le surajustement. En utilisant une méthode de descente de gradient, le modèle XGBoost ajuste les paramètres des arbres pour optimiser cette fonction, tout en appliquant des techniques de régularisation telles que la réduction du nombre d'arbres, la limitation de leur profondeur et la pénalisation des poids des feuilles.
3. **Combinaison des arbres :** Une fois que tous les arbres faibles sont construits, le modèle XGBoost les combine pour créer un modèle puissant. Cette fusion s'effectue en agrégeant les prédictions de chaque arbre, en régression on utilise une moyenne pondérée.
4. **Finalisation du modèle et prédictions :** Le modèle entraîné est prêt à être utilisé pour faire des prédictions sur de nouvelles données en passant ces données à travers l'ensemble des arbres construits.

2.4.2 L'algorithme de l'Extreme Gradient Boosting

L'objectif de ce modèle est de simplifier les arbres en pénalisant les poids de leurs feuilles afin de prévenir le surajustement[9].

1. Soit le modèle XGBoost, composé d'un ensemble de K arbres de régression (CART) prédit, avec un ensemble d'entrainement de n observations et m variables
   $\mathscr{D} = \{(x_i, y_i), i \in \{1, ..., n\}, x_i \in \mathbb{R}^m, y_i \in \mathbb{R}\}$.
   Le modèle additif d'agrégation d'arbres est le suivant :

$$\hat{y}_i = \sum_{k=1}^{\mathrm{K}} f_k(x_i), \quad f_k \in \mathscr{F}. \tag{2.5}$$

   Où $\mathscr{F}$ est l'ensemble des arbres de régression CART, et chaque f_k représente un arbre de nombre de feuilles T, de poids de feuilles w.

   En posant $\mathrm{I}_j = \{i | q(x_i) = j\}$ l'ensemble des individus qui respectent les conditions conduisant à la $j^{\text{è}me}$ feuille, et q étant la structure de l'arbre. Alors la prédiction d'un arbre f_k est obtenue par :

$$f_k(x_i) = w_{q(x_i)} = w_j, \quad i \in \mathrm{I}_j. \tag{2.6}$$

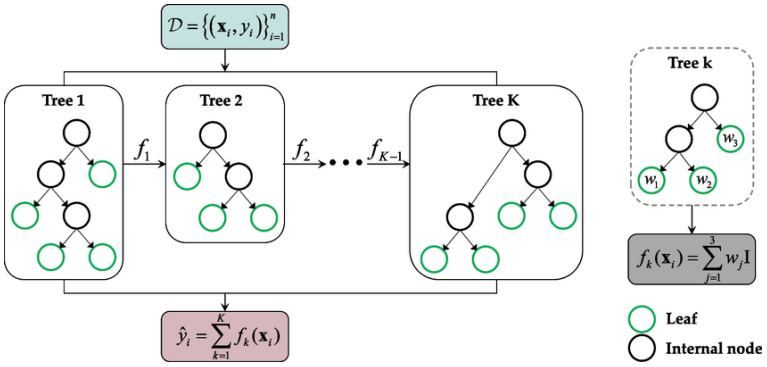


FIGURE 2.13 – L'algorithme de XGBoost.

2. La fonction objectif Φ se décompose en la somme de la fonction de perte j et d'un terme de régularisation Ω :

$$\Phi(\hat{f}(x)) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{\mathrm{K}} \Omega(f_k). \tag{2.7}$$

En prenant la MSE comme fonction de perte l, on aura :

$$\sum_{i=1}^{n} l(y_i, \hat{y}_i) = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{2.8}$$

Le terme de régularisation qui vise à limiter le sur-apprentissage qui se produit lors de la construction séquentielle des arbres de régression, est le suivant :

$$\Omega(f) = \gamma \mathrm{T} + \frac{1}{2}\lambda \sum_{j=1}^{\mathrm{T}} w_j^2. \tag{2.9}$$

O*ù*, γ contrôle l'importance de la régularisation ; plus gamma est élevé, plus la régularisation est forte. Cela entraîne une pression accrue sur la réduction des poids.
Et λ contrôle la pénalisation des poids des feuilles.

On commence avec un arbre de profondeur nulle et on ajoute des nœuds de manière récursive. Dont l'objectif est de minimiser la perte de ce que nous avons déjà appris.

$$\begin{array}{lllll}
\hat{y}_i^{(0)} & = & 0 & & \\
\hat{y}_i^{(1)} & = & f_1(x_i) & = & \hat{y}_i^{(0)} + f_1(x_i) \\
\hat{y}_i^{(2)} & = & f_1(x_i) + f_2(x_i) & = & \hat{y}_i^{(1)} + f_2(x_i) \\
\vdots & \vdots & \vdots & & \\
\hat{y}_i^{(t)} & = & \sum_{k=1}^{t} f_k(x_i) & = & \hat{y}_i^{(t-1)} + f_t(x_i).
\end{array}$$

Avec t le nombre d'itérations.

La fonction objectif devient :

$$\Phi^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \tag{2.10}$$

$$\Phi^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + c. \tag{2.11}$$

O*ù* c une constante.

Maintenant, d'après la proposition de Chen[9], qui est l'utilisation d'un développement de Taylor d'ordre 2, pour simplifier la minimisation de l'objectif.

$$\Phi^{(t)} = \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + c. \tag{2.12}$$

O*ù* $g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ et $h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$ sont respectivement les dérivées partielles du premier et du second ordre de la fonction de perte.

Simplifier et supprimer la constante par rapport à f_t :

$$\tilde{\Phi}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t). \tag{2.13}$$

En remplaçant $\Omega(f_t)$ et f_t par leurs formules, l'objectif devient :

$$\begin{aligned} \tilde{\Phi}^{(t)} &\approx \sum_{i=1}^{n} \left[g_i w_{q(x_i)} + \frac{1}{2} h_i w^2_{q(x_i)} \right] + \gamma \mathrm{T} + \frac{1}{2} \lambda \sum_{j=1}^{\mathrm{T}} w_j^2. && (2.14) \\ &= \sum_{j=1}^{\mathrm{T}} \left[(\sum_{i \in \mathrm{I}_j} g_i) w_j + \frac{1}{2} (\sum_{i \in \mathrm{I}_j} h_i + \lambda) w_j^2 \right] + \gamma \mathrm{T}. && (2.15) \end{aligned}$$

En posant

$$\begin{aligned} \mathrm{G}_j &= \sum_{i \in \mathrm{I}_j} g_i. \\ \mathrm{H}_j &= \sum_{i \in \mathrm{I}_j} h_i. \end{aligned}$$

On obtient la nouvelle fonction objectifve, ci-dessous :

$$\tilde{\Phi}^{(t)} = \sum_{j=1}^{\mathrm{T}} \left[\mathrm{G}_j w_j + \frac{1}{2} (\mathrm{H}_j + \lambda) w_j^2 \right] + \gamma \mathrm{T}. \tag{2.16}$$

Dans cette équation, les w_j sont indépendants les uns par rapport aux autres.
Pour une structure $q(x)$ donnée, on obtient le vecteur du poids optimal w_j^* d'une feuille j en minimisant $\tilde{\Phi}^{(t)}$.

Pour j fixé, on pose χ qui est à minimiser :

$$\begin{aligned} \chi(w) &= w(\sum_{i \in \mathrm{I}_j} g_i) + \frac{1}{2} w^2 (\sum_{i \in \mathrm{I}_j} h_i + \lambda). && (2.17) \\ \frac{\partial \chi}{\partial w}(w) &= \sum_{i \in \mathrm{I}_j} g_i + w(\sum_{i \in \mathrm{I}_j} h_i + \lambda). && (2.18) \end{aligned}$$

Et

$$\frac{\partial^2 \chi}{\partial w^2}(w) = \sum_{i \in \mathrm{I}_j} h_i + \lambda. \tag{2.19}$$

Donc χ est convexe car $\lambda > 0$ et que $\sum_{i \in \mathrm{I}_j} h_i \geq 0$ car l est supposée convexe.

Le minimum de χ est donc atteint pour w^*, tel que :

$$\frac{\partial \chi}{\partial w}(w^*) = 0. \tag{2.20}$$

Donc,

$$\sum_{i \in \mathrm{I}_j} g_i + w^* (\sum_{i \in \mathrm{I}_j} h_i + \lambda) = 0. \tag{2.21}$$

Alors le vecteur du poids optimal w^* est le suivant :

$$w^* = -\frac{\mathrm{G}_j}{\mathrm{H}_j + \lambda}. \tag{2.22}$$

Et la fonction objective devient :

$$\Phi^* = -\frac{1}{2} \sum_{j=1}^{\mathrm{T}} \frac{\mathrm{G}_j^2}{\mathrm{H}_j + \lambda} + \gamma \mathrm{T}. \tag{2.23}$$

Cette fonction objective $\tilde{\Phi}^{(t)}$ permet de mesurer l'efficacité de la structure q. Plus $\tilde{\Phi}^{(t)}$ est faible, plus q est bonne. Cependant, il est impossible de comparer toutes les structures d'arbres possibles, étant donné qu'il en existe une infinité.

Lorsqu'un nœud est ajouté à une feuille de l'arbre précédent, elle est alors divisée en deux. Et on crée de nouvelles feuilles qui permettent de mieux capturer la structure des données et donc de réduire la fonction objectif, améliorant ainsi la performance du modèle.
La diminution de la fonction objectif est alors calculée comme suit :

$$\Phi_{split} = \frac{1}{2} \Big[\frac{\mathrm{G}_\mathrm{L}^2}{\mathrm{H}_\mathrm{L} + \lambda} + \frac{\mathrm{G}_\mathrm{L}^2}{\mathrm{H}_\mathrm{L} + \lambda} + \frac{(\mathrm{G}_\mathrm{L} + \mathrm{G}_\mathrm{R})^2}{\mathrm{H}_\mathrm{L} + \mathrm{H}_\mathrm{R} + \lambda} \Big] - \gamma. \tag{2.24}$$

Ce processus se répète jusqu'à ce qu'un critère d'arrêt soit atteint ou que la performance du modèle soit satisfaisante. En somme, cette méthode itérative offre une manière efficace de trouver une approximation de la structure optimale de l'arbre de décision.

2.4.3 Principaux paramètres et mécanismes d'optimisation

Les hyper-paramètres ci-dessous sont ajustés pour optimiser le modèle[15].

1. **Nombre d'arbres (n_estimators) :** Un nombre plus élevé d'arbres peut améliorer les performances du modèle, mais cela peut également augmenter le temps d'entraînement et le risque de surapprentissage.

2. **Profondeur maximale de l'arbre (max_depth) :** Une profondeur plus élevée peut conduire à un modèle plus complexe, ce qui peut entraîner un surapprentissage.
3. **Taux d'apprentissage (learning_rate) :** Il contrôle la vitesse à laquelle le modèle apprend. Un taux d'apprentissage plus faible peut conduire à une convergence plus lente mais à des performances potentiellement meilleures.
4. **Régularisation (reg_lambda) :** Le terme de régularisation L2 qui pénalise les poids des feuilles des arbres pour éviter le surapprentissage.
5. **Échantillonnage aléatoire (subsample) :** Taille des sous-échantillons d'observations pour chaque nouvelle construction d'arbre.
6. **Gamma :** Seuil minimal de réduction de l'erreur.
7. **Fraction de fonctionnalités (Colsample_bytree)** Taille des sous-échantillons de variables considérées pour chaque nouvelle construction d'arbre.
8. **Min_child_weight :** Nombre minimum d'observations par feuille.
9. **Arrêt anticipé (early_stopping_rounds) :** Mécanisme permettant d'arrêter l'entraînement lorsque la performance sur l'ensemble de validation cesse de s'améliorer, afin d'éviter le surajustement.

2.4.4 Les avantages et inconvénients

XGBoost présente plusieurs avantages et inconvénients[30] :

2.4.4.1 Avantages

Haute performance : Le modèle XGBoost est connu pour sa performance en termes de précision et de vitesse, ce qui en fait un choix populaire pour les problèmes de classification et de régression. Avec des résultats de haute qualité dans diverses tâches d'apprentissage automatique.

Capacité à gérer des ensembles de données massifs : Il peut s'adapter pour traiter des ensembles de données de différentes tailles et complexités, grâce à ses algorithmes optimisés.

Optimisation des hyperparamètres : Le modèle XGBoost offre une variété d'hyperparamètres ajustables, permettant ainsi d'optimiser ses performances, le rendant ainsi flexible et adaptable.

Gestion automatique des valeurs manquantes : Ce modèle est capable de traiter automatiquement les valeurs manquantes dans les données, ce qui diminue la nécessité de prétraiter les données.

Interprétabilité des modèles : Le modèle XGBoost offre des outils pour évaluer l'importance des fonctionnalités, ce qui permet de mieux comprendre les variables les plus significatives pour les prédictions.

2.4.4.2 Inconvénients

Sensibilité aux paramètres : Le modèle XGBoost possède de nombreux hyperparamètres ajustables, ce qui souligne l'importance de trouver le bon réglage pour optimiser les performances. Cela peut demander du temps et nécessite une expertise.

Surajustement : Il peut être sujet au surapprentissage, comme tout modèle d'apprentissage automatique. Cela se produit notamment lorsque les paramètres ne sont pas correctement réglés ou lorsque le modèle est entraîné sur des petits ensembles de données, ou encore lorsque trop d'arbres sont utilisés.

Exigences en termes de puissance de calcul : Le modèle XGBoost peut exiger des calculs importants, surtout lorsqu'il est appliqué à des ensembles de données massifs ou lorsqu'il est associé à une exploration complète des hyperparamètres.

Temps de calcul : Ce modèle est généralement rapide, mais l'entraînement de modèles sur de très grands ensembles de données peut prendre du temps, en particulier avec des configurations complexes ou des hyperparamètres mal réglés.

Exigences en matière de mémoire : Il peut nécessiter une quantité importante de mémoire, surtout lorsqu'il est utilisé avec de vastes ensembles de données, ce qui peut le rendre moins approprié pour les systèmes ayant des ressources de mémoire limitées.

2.5 Light Gradient Boosting Machine (LightGBM)

Light Gradient Boosting Machine, ou LightGBM, est un modèle d'apprentissage automatique populaire utilisé dans les problèmes de classification, de régression et de classement[36]. Développé par Microsoft et devenu open source en 2017, ce modèle de boosting supervisé utilise un ensemble d'arbres de décision. Il est devenu largement utilisé dans l'industrie et la recherche en raison de sa grande efficacité et de ses performances élevées[1].

Le modèle LightGBM a été développé pour surmonter certains des inconvénients de XGBoost, en améliorant notamment l'efficacité de l'apprentissage et en réduisant le temps de traitement. Cette amélioration est rendue possible par l'utilisation d'une méthode de division par feuille, connue sous le nom de leaf-wise (feuille par feuille), et une optimisation efficace du processus d'apprentissage. Le modèle LightGBM choisit la division qui réduit le plus l'erreur à chaque étape, ce qui peut conduire à des arbres plus profonds mais plus étroits, selon une fonction de perte spécifique. Contrairement au modèle XGBoost qui utilise une approche de division level-wise (niveau par niveau), qui divise l'arbre de manière équilibrée à chaque niveau, favorisant des arbres plus larges mais moins profonds[41]. Cependant, le modèle LightGBM se distingue par sa capacité à traiter efficacement de grands ensembles de données et à effectuer des apprentissages rapidement, tout en maintenant de bonnes performances prédictives[11].

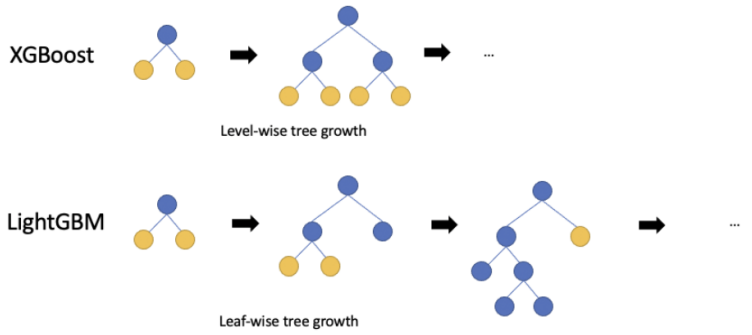


FIGURE 2.14 – La différence entre XGBoost et LightGBM.

2.5.1 Fonctionnement de l'algorithme LightGBM

Le modèle LightGBM fonctionne de la même manière que XGBoost, mais avec des caractéristiques avancées et uniques. Cependant, on va voir comment fonctionne le modèle de boosting LightGBM et en quoi il diffère de l'autre modèle[21] :

1. **Construction des arbres de décision :** Le modèle LightGBM commence par la construction d'un ensemble d'arbres de décision faibles, où chaque arbre est construit séquentiellement, en utilisant la méthode innovante appelée "Leaf-wise tree growth".
2. **Échantillonnage unilatéral basé sur le gradient (Gradient-based One-Side Sampling (GOSS)) :** Le modèle LightGBM utilise Gradient-based One-Side Sampling, pour prioriser les échantillons d'apprentissage avec de grandes erreurs. Plus la valeur du gradient est élevée, plus l'erreur est grandes.
   Dans ce cas, GOSS conserve tous les exemples présentant des gradients (erreur) élevés et échantillonne de manière aléatoire les exemples présentant des gradients faibles. Cela permet d'améliorer l'efficacité de l'apprentissage.

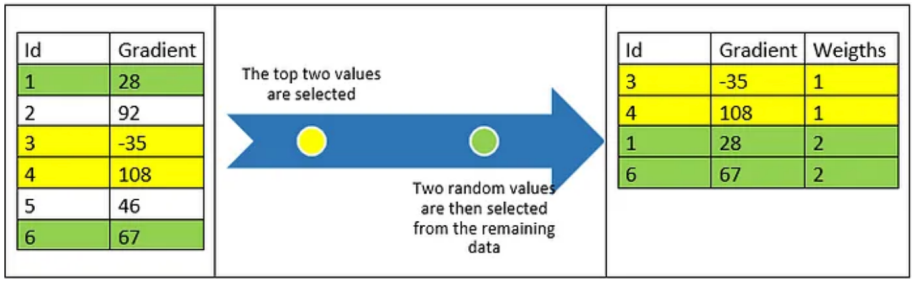


FIGURE 2.15 – Échantillonnage unilatéral basé sur le gradient.

3. **Utilisation de l'algorithme des histogrammes :** Pendant la construction de chaque arbre, le modèle utilise un histogramme pour représenter les valeurs des caractéristiques. L'histogramme discrétise les valeurs continues en plusieurs intervalles, facilitant ainsi les décisions de division.

Salary (k)	Bins
30	30-50
40	
50	
60	51-80
70	
80	

FIGURE 2.16 – L'algorithme des histogrammes.

Il est nécessaire de créer des bins pour chaque élément et de choisir le bin le plus efficace pour réduire les pertes.

4. **Sélection des divisions optimales :** Il compare les histogrammes des nœuds enfants avec celui du nœud parent pour déterminer la prochaine division. La fonction de perte est également évaluée pour chaque candidat de division, et celle qui entraîne la plus grande réduction de la perte est choisie.
5. **Regroupement de fonctionnalités exclusives (Exclusive Feature Bundling (EFB)) :** La méthode Exclusive Feature Bundling (EFB) utilisée par LightGBM est une technique d'optimisation visant à réduire la dimensionnalité des données tout en conservant autant d'informations que possible. Elle regroupe des caractéristiques exclusives entre elles, c'est-à-dire des caractéristiques qui ne prennent jamais de valeurs non nulles simultanément dans le même échantillon, afin de réduire la complexité du modèle et d'accélérer le processus d'apprentissage.

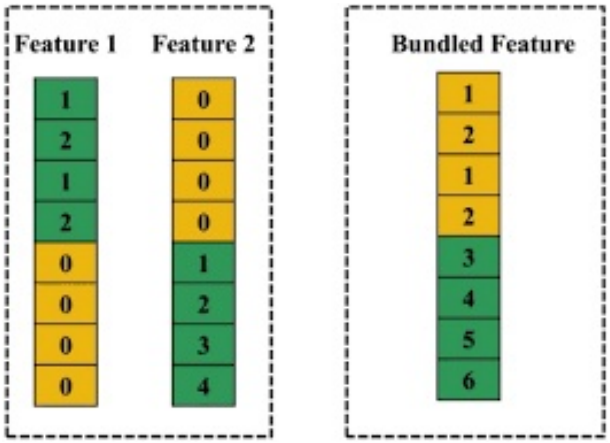


FIGURE 2.17 – Fonctionnement du regroupement de fonctionnalités exclusives.

6. **Finalisation du modèle et prédictions :** Après que tout les arbres faibles sont agrégées, le modèle LightGBM est prêt à faire des prédictions sur de nouvelles données. Les méthodes GOSS, l'histogramme et la fonction de perte contribuent à rendre ce processus rapide et efficace en réduisant la taille des données et la complexité du modèle.

2.5.2 L'algorithme de Light Gradient Boosting Machine

Les deux principales différences avec le modèle XGBoost se trouvent dans les méthodes innovantes utilisées par LightGBM, qui sont conçues pour accélérer le processus de calcul tout en préservant des performances élevées[11] :

2.5.2.1 Échantillonnage unilatéral basé sur le gradient

Cette méthode initiale GOSS, qui réduit le nombre d'observations considérées pour diviser le nœud optimal en calculant le gain de variance. Dont les a premières observations triées par le gradient décroissant sont conservées (ensemble A), puis un échantillon de b observations du reste est sélectionné aléatoirement (ensemble B), dont $B \subseteq A^c$. Pour éviter de trop perturber la distribution des observations, les gradients échantillonnées sont pondérés par un facteur $\frac{1-a}{b}$ lors du calcul du gain de variance $\tilde{V}_j$.

$$\tilde{V}_j(d) = \frac{1}{n}\Big(\frac{(\sum_{x_i \in A_L} g_i + \frac{1-a}{b}\sum_{x_i \in B_L} g_i)^2}{n_L^j(d)} + \frac{(\sum_{x_i \in A_R} g_i + \frac{1-a}{b}\sum_{x_i \in B_R} g_i)^2}{n_R^j(d)}\Big). \tag{2.25}$$

Avec,
$A_L = \{x_i \in A : x_{ij} \leq d\}$, $A_R = \{x_i \in A : x_{ij} > d\}$, $B_L = \{x_i \in B : x_{ij} \leq d\}$, $B_R = \{x_i \in B : x_{ij} > d\}$.

D'o*ù*, g_i est le gradient, A_L et B_L représentent respectivement les sous-ensembles des parties A et B au point de coupure d pour la caractéristique j dans la feuille gauche, A_R et B_R le sont pour le côté droit. Quant à n_L^j et n_R^j, ils désignent les nombres d'observations respectifs.

Un gain de variance élevé indique que la division envisagée permet de diminuer significativement la variation des valeurs cibles. Cela implique que les sous-groupes résultants sont plus similaires entre eux en termes de valeurs cibles. Ainsi, cette division est jugée bénéfique pour construire l'arbre de décision.

Une borne supérieure théorique de l'erreur est connue, ce qui suggère que lorsque le nombre d'observations est élevé et que le seuil choisi pour la sélection des données n'introduit pas de déséquilibre important, l'approximation est de haute qualité.
L'efficacité de l'erreur de généralisation est significative si l'erreur initiale de GOSS est déjà faible. Dans le cas contraire, l'utilisation d'un échantillonnage classique peut être bénéfique pour améliorer l'apprentissage.

2.5.2.2 Regroupement de fonctionnalités exclusives

En regroupant les variables exclusives dans un "bundle", l'approche Exclusive Feature Bundling (EFB) permet de réduire la complexité des données tout en conservant le plus d'informations pertinentes possible. Cela rend l'analyse des modèles d'apprentissage automatique plus facile en réduisant le nombre de variables à considérer.

Alors l'EFB suit ces étapes :

1. **Exclusivité des fonctionnalités** On commence par identifier les variables qui ne prennent jamais de valeurs non nulles simultanément dans le même échantillon.
2. **Regroupement des variables :** Ensuite, les variables exclusives sont rassemblées dans un "bundle".
3. **Itération :** Ce processus de regroupement des variables est répété de manière itérative jusqu'à ce qu'un critère d'arrêt prédéterminé soit atteint, comme le nombre total de "bundles" créés ou la réduction de la corrélation moyenne entre les variables.

2.5.3 Principaux paramètres et mécanismes d'optimisation

Light Gradient Boosting Machine ajuste ces hyper-paramètres, pour optimiser le modèle[41] :

1. **Nombre d'arbres (n_estimators)**
2. **Taux d'apprentissage (learning_rate)**
3. **Régularisation (reg_lambda)**
4. **Échantillonnage aléatoire (subsample)**
5. **Gamma**
6. **Fraction de fonctionnalités (Colsample_bytree)**
7. **Nombre de feuilles (num_leaves) :** Nombre maximum de feuilles dans un arbre.
8. **Gain de division minimal (min_split_gain) :** si le gain obtenu en divisant un noeud est inférieur à cette valeur spécifiée, alors ce noeud ne sera pas divisé.

2.5.4 Les avantages et inconvénients

LightGBM présente plusieurs avantages et inconvénients[36] :

2.5.4.1 Avantages

Haute performance : Le modèle LightGBM est rapide et efficace, ce qui le rend idéal pour les gros ensembles de données.

Traitement efficace des données volumineuses : Il peut gérer de grandes quantités de données grâce à son algorithme de partitionnement en feuilles, ce qui simplifie les calculs.

Utilisation minimale de la mémoire : Contrairement à d'autres modèles similaires, le modèle LightGBM utilise peu de mémoire, ce qui le rend adapté aux environnements avec des ressources limitées.

Précision élevée : Il offre généralement de bonnes performances de prédiction grâce à son algorithme optimisé.

Personnalisation des paramètres : Le modèle LightGBM propose de nombreux paramètres réglables, permettant aux utilisateurs d'ajuster le modèle selon leurs besoins.

2.5.4.2 Inconvénients

Risque de surajustement : Le modèle LightGBM peut être sensible au surajustement, surtout avec de petits ensembles de données ou des paramètres mal réglés.

Difficulté d'interprétation : Les modèles créés par LightGBM peuvent être complexes, rendant leur interprétation difficile pour les utilisateurs moins expérimentés.

Besoin d'optimisation des paramètres : Pour obtenir les meilleures performances, le modèle LightGBM nécessite souvent une optimisation minutieuse des paramètres, ce qui prend beaucoup de temps et nécessite des ressources supplémentaires.

2.6 Categorical Boosting (CatBoost)

Le modèle CatBoost, également connu sous le nom de Categorical Boosting, a été développé en 2018 par Prokhorenkova[34] et amélioré par un modèle de Boosting des arbres de décision binaire. Il introduit deux innovations majeures par rapport aux modèles traditionnelles (XGBoost et LightGBM) : le boosting ordonné et une nouvelle méthode d'encodage des variables catégorielles appelée statistique cible ordonnée (Target Statistic (TS)). Ces avancées permettent de mieux gérer les caractéristiques catégorielles et d'éviter le surapprentissage, ce qui économise les ressources de calcul. Cependant, ce modèle vise à résoudre les problèmes de généralisation causés par le fait que le gradient calculé à chaque itération utilise les mêmes valeurs de la variable cible sur lesquelles il a été optimisé[2].

Ce modèle supervisé est reconnu pour sa robustesse, sa polyvalence et sa rapidité de prédiction. Le modèle CatBoost peut gérer à la fois les problèmes de classification et de régression. Il repose sur l'apprentissage ensembliste complexe et utilise le partitionnement équilibré des données dans les arbres de décision pour construire des classificateurs faibles. Grâce à ce partitionnement, CatBoost crée un modèle indépendant pour chaque échantillon, évitant ainsi tout biais de prédiction causé par des fuites d'information pendant le processus d'entraînement, ce qui améliore la précision des prédictions. De plus, en raison des caractéristiques structurelles des arbres de décision symétriques, CatBoost réduit efficacement le risque de surapprentissage et améliore considérablement la vitesse de prédiction[2].

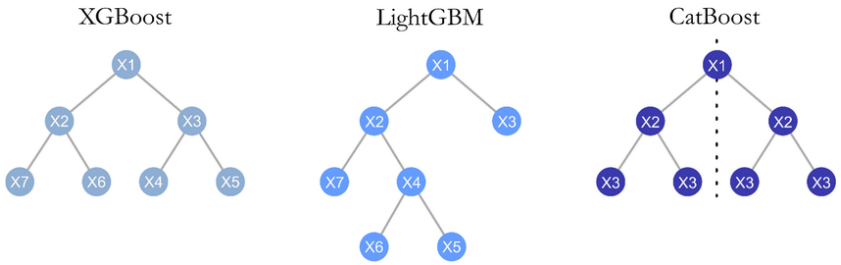


FIGURE 2.18 – La différence entre XGBoost, LightGBM et CatBoost.

2.6.1 Fonctionnement de l'algorithme CatBoost

Contrairement à d'autres modèles de boosting, CatBoost peut traiter les caractéristiques catégorielles directement sans nécessiter de prétraitement supplémentaire tel que le codage Greedy Target Statistic. De plus, le modèle CatBoost utilise une technique appelée ordered boosting pour prévenir le surajustement du modèle, ce qui améliore la généralisation et la précision des prédictions[2].

1. **Division aléatoire des sous-ensembles de données :** Le modèle CatBoost divise les données en sous-ensembles de manière aléatoire pour faciliter l'entraînement efficace et éviter le surajustement.
2. **Codage des étiquettes :** Le modèle convertit les étiquettes en nombres entiers pour représenter numériquement différentes classes ou catégories.
3. **Transformation des caractéristiques catégorielles :** Il transforme les caractéristiques catégorielles en valeurs numériques à l'aide de techniques telles que le codage des statistiques cibles ordonnées (TS) pour gérer efficacement les données catégorielles.
4. **Construction des arbres de décision :** Le modèle CatBoost utilise le boost ordonné pour créer des ensembles d'arbres de décision de manière séquentielle à l'aide d'ensembles de données triés, réduisant ainsi les biais et améliorant la précision des prévisions.
5. **Optimisation de la fonction de perte :** Le modèle CatBoost vise à minimiser une fonction de perte spécifique pendant l'entraînement en utilisant la descente en pente pour ajuster les paramètres du modèle.
6. **Prédiction finale :** les prédictions individuelles de chaque arbre sont combinées pour générer le résultat de prédiction final.

2.6.2 L'algorithme de Categorical Boosting

Le modèle CatBoost vise à améliorer les performances prédictives des modèles d'apprentissage automatique, en particulier en ce qui concerne la gestion efficace des variables catégorielles, en suivant ces techniques[34] :

2.6.2.1 Greedy Target Statistic

Greedy Target Statistic une façon efficace et efficiente de traiter une caractéristique catégorielle x_j^i est de la remplacer par une caractéristique numérique $\hat{x}_j^i$, égale à une statistique cible estimée (TS). Généralement, cela consiste à estimer la cible attendue y conditionnée par la catégorie : $\hat{x}_j^i \approx \mathbb{E}(y|x^i = x_j^i)$, l'approche la plus simple c'est de prendre la moyenne des valeurs de y sur les exemples d'entraînement qui appartiennent à la même catégorie x_j^i. Cette estimation peut être instable pour les catégories de faible fréquence, et pour résoudre ce problème, on lisse souvent en utilisant une distribution a priori p, comme suit :

$$\hat{x}_j^i = \frac{\sum_{k=1}^{n} \mathbb{1}_{x_k^i = x_j^i} y_k + ap}{\sum_{k=1}^{n} \mathbb{1}_{x_k^i = x_j^i} + a}. \tag{2.26}$$

où n est le nombre d'observations sur la base d'apprentissage, $a > 0$ est un hyper-paramètre à fixer, tel que plus la valeur de a est élevée, plus cet estimateur est régularisé, et $p = \frac{\sum_{j=1}^{n} \mathbb{1}_{y_j=1}}{n}$ est la valeur moyenne de toutes les valeurs cibles.

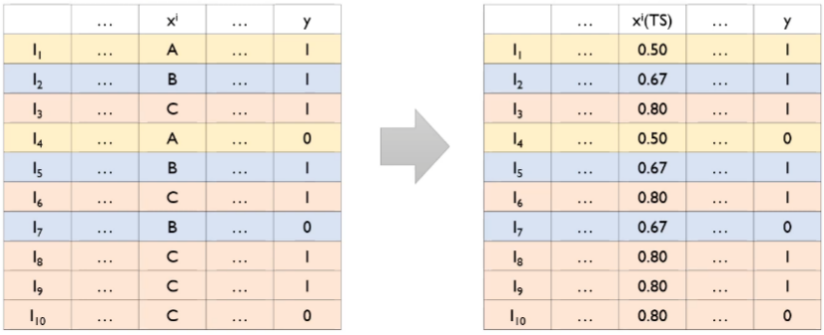


FIGURE 2.19 – Remplacer les caractéristiques catégorielles par des caractéristiques numériques.

— Si les catégories présentes dans la base d'apprentissage sont aussi présentes dans la base de test, alors le codage des catégories est appliqué de la même manière.
— Le problème de cette approche est la fuite cible qu'on appelle aussi "target leakage", qui est un décalage conditionnelle.
Par exemple, si les catégories sont uniques, alors la catégorie dans l'ensemble de test est $\hat{x}_j^i = p$.
Ainsi, l'ensemble d'apprentissage, sa statistique $\hat{x}_j^i$ devient :

$$\hat{x}_j^i = \frac{y_j + ap}{1 + a}. \tag{2.27}$$

Et si la probabilité $\mathbb{P}(y = 1|x^i = \text{A}) = 0.5$ pour toutes les catégories, alors, il suffit d'effectuer une seule division dont le seuil est $d = \frac{0.5+ap}{1+a}$, où toutes les observations sont regroupées dans un même nœud, et la précision du modèle vaut 0.5.

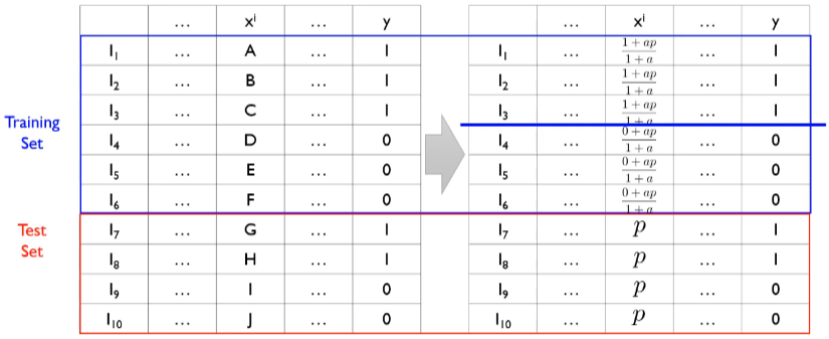


FIGURE 2.20 – Exemple de la fuite cible.

Plusieurs approches ont été suggérées pour aborder ce probléme. Elles consistent principalement à ne pas inclure l'observation j de l'estimateur pour x_j^i et à considérer un sous-ensemble $\mathrm{D}_j \subset \mathrm{D} \setminus x_j$, la statistique devient :

$$\hat{x}_j^i = \frac{\sum_{x_k \in \mathrm{D}_j} \mathbb{1}_{x_k^i = x_j^i} y_k + ap}{\sum_{x_k \in \mathrm{D}_j} \mathbb{1}_{x_k^i = x_j^i} + a}. \tag{2.28}$$

2.6.2.2 Ordered boosting

Cet algorithme, appelé Boosting ordonné, est utilisé dans le modèle CatBoost pour résoudre le problème de décalage de prédiction.

1. **Entrée :** L'algorithme prend en entrée un ensemble de données d'apprentissage $\{(x_j, y_j)\}_{j=1}^n$, où n est le nombre total de l'ensemble d'apprentissage, et I le nombre d'itérations pour l'apprentissage.
2. **Permutation aléatoire :** Une permutation aléatoire σ de l'ensemble d'indices $[1, n]$ est générée. Cette permutation est utilisée pour obtenir une variété de perspectives sur les données lors de l'apprentissage des modèles.
3. **Initialisation :** Les modèles de support M_j sont initialisés à zéro pour $j = [1, n]$.
4. (a) Pour chaque permutation j le calcul des résidus est effectué :

$$r_j = y_i - \mathrm{M}_{\sigma_{(j)}-1}(x_j). \tag{2.29}$$

   (b) Un modèle partiel $\triangle \mathrm{M}_j$ est entraîné en utilisant un sous-ensemble différent de données $\{(x_k, y_k)\}_{k=1}^n$ jusqu'à atteindre la position j de la permutation actuelle.

   (c) Chaque modèle M_j est mis à jour en ajoutant le modèle partiel correspondant $\triangle \mathrm{M}_j$.

$$\mathrm{M}_j = \mathrm{M}_j + \triangle \mathrm{M}_j. \tag{2.30}$$

Ce processus itératif se poursuit jusqu'à ce que le niveau de précision souhaité soit atteint.

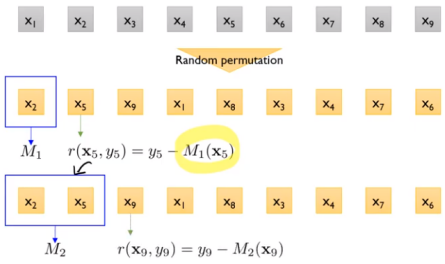


FIGURE 2.21 – Ordered boosting.

2.6.3 Principaux paramètres et mécanismes d'optimisation

CatBoost ajustes ces hyper-paramètres, pour optimiser le modèle[22] :

1. **Nombre d'arbres (n_estimators)**
2. **Taux d'apprentissage (learning_rate)**
3. **Régularisation (reg_lambda)**
4. **Profondeur maximale de l'arbre (max_depth)**
5. **Échantillonnage aléatoire (subsample)**
6. **Nombre minimal d'échantillons dans une feuille (min_data_in_leaf)** Il contrôle la profondeur de l'arbre.

2.6.4 Les avantages et inconvénients

CatBoost présente plusieurs avantages et inconvénients[17] :

2.6.4.1 Avantages

Gestion des variables catégorielles : Le modèle CatBoost gère automatiquement les fonctionnalités catégorielles, ce qui réduit la complexité du prétraitement approfondi des données.

Haute performance : Le modèle CatBoost est reconnu pour sa vitesse d'exécution et sa capacité à gérer des ensembles de données volumineux.

Bonne performance prédictive : Il produit généralement des modèles avec de bonnes performances prédictives, en particulier lorsqu'il est bien ajusté.

Généralisation améliorée : Le modèle CatBoost introduit le boost ordonné et les statistiques cibles ordonnées, résolvant ainsi les problèmes de généralisation présents dans les modèles de boost traditionnels tels que XGBoost et LightGBM.

Robustesse : La gestion des caractéristiques catégorielles par CatBoost améliore la robustesse du modèle et la précision prédictive.

2.6.4.2 Inconvénients

Sensibilité aux valeurs manquantes : les performances de CatBoost peuvent être affectées lorsqu'il traite des valeurs manquantes, ce qui peut entraîner une sous-performance par rapport à d'autres modèles.

Complexité informatique : Le modèle CatBoost peut nécessiter beaucoup de calculs, en particulier pour les grands ensembles de données, en raison de ses algorithmes avancés et de sa gestion des variables catégorielles.

Réglage des paramètres : L'optimisation des hyperparamètres de CatBoost peut être complexe, car elle exige une compréhension approfondie du modèle et de son influence sur les performances du modèle.

Complexité des modèles : Le modèle CatBoost génére Les modèles qui peuvent être complexes, ce qui rend leur interprétation plus difficile par rapport aux modèles plus simples.

2.7 Conclusion

Dans ce chapitre nous avons exploré plusieurs modèles de prévision de l'apprentissage automatique, en commençant par les réseaux de neurones complexes. Ensuite, par les approches les plus simples comme les modèles ensemblistes de boosting (XGBoost, LightGBM et CatBoost). Chaque modèle offre ses propres avantages qui conduit à des performances améliorées dans la prédiction de données complexes.

CHAPITRE 3

APPLICATIONS DES MODÈLES DE PRÉDICTION DE LA VOLATILITÉ DANS DIVERS SECTEURS ÉCONOMIQUES

3.1 Introduction

Ce chapitre présente une analyse comparative des performances de différents modèles de prédiction de la volatilité. L'étude originale de Curtis Nybo [29] comparait les modèles GARCH et ANN, deux approches traditionnelles pour la modélisation des séries temporelles financières dans différents secteurs économiques. Dans notre étude, nous avons élargi cette comparaison en ajoutant trois modèles de boosting modernes : XGBoost, LightGBM et CatBoost. Cette inclusion vise à évaluer si ces modèles de boosting, reconnus pour leur flexibilité et leur performance sur des données complexes, peuvent surpasser les modèles traditionnels dans la prédiction de la volatilité pour différents secteurs, tels que la consommation durable, la santé, la technologie, l'industrie manufacturière et d'autres secteurs divers.

Pour chaque secteur, nous avons déterminé le modèle le plus performant en utilisant des métriques d'évaluation telles que RMSE, MAE et MSE. La comparaison se base principalement sur la métrique RMSE pour identifier le modèle offrant les prévisions les plus précises.

3.2 Déscription des données

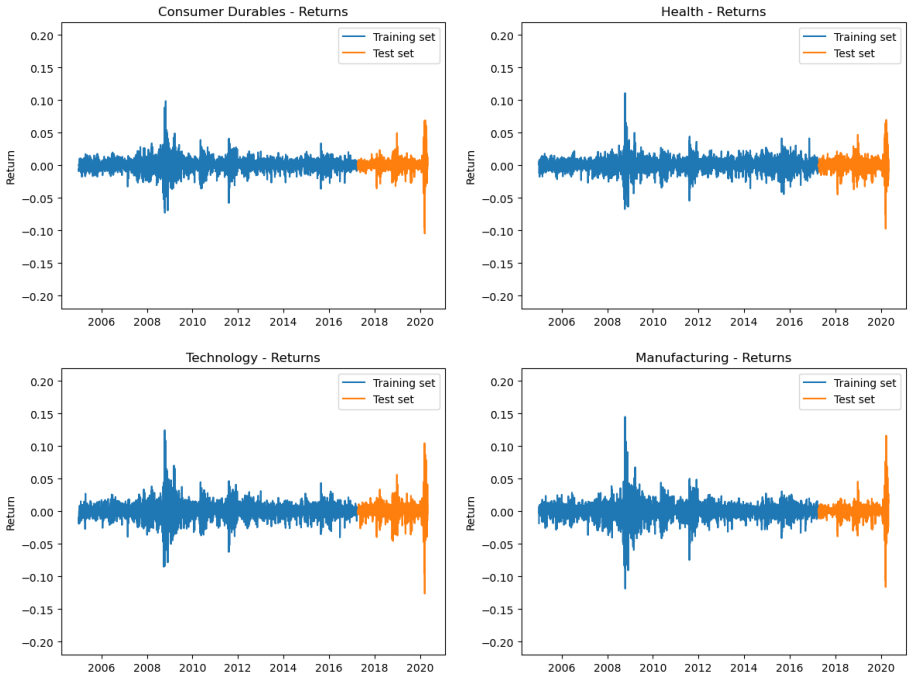
Pour cette étude, on a utilisé des données provenant de cinq portefeuilles industriels de la bibliothèque de données française Kenneth R. Ces données comprennent les rendements quotidiens des actions des bourses New York Stock Exchange (NYSE), American Stock Exchange (AMEX) et National Association of Securities Dealers Automated Quotations (NASDAQ), couvrant la période du 3 janvier 2005 au 30 avril 2020, soit un total de 3 858 observations. Les données sont classées en cinq secteurs principaux selon les codes de la classification industrielle

type (Standard Industrial Classification (SIC)) : biens de consommation durables, santé, technologie, industrie manufacturière et autres secteurs.

Les rendements quotidiens ont été calculés pour chaque secteur, capturant des périodes de forte volatilité telles que la crise financière de 2008, la correction du marché en 2011-2012, la chute des prix du pétrole en 2016, et plus récemment, l'incertitude liée à la pandémie de Covid-19 et une autre chute des prix du pétrole début 2020. Ces événements sont clairement visibles dans les graphiques des rendements quotidiens, montrant des fluctuations significatives et des périodes de turbulences sur le marché.

3.2.1 Préparation des données

Pour préparer les données, on a d'abord éliminé les valeurs manquantes et identifié les valeurs aberrantes pour assurer la qualité et la précision des données. Ensuite, on a converti les dates en format datetime pour faciliter leur manipulation et on les a définies comme index du DataFrame pour permettre une analyse temporelle plus efficace. Puis, on a calculé la volatilité en utilisant les rendements au carré, une méthode couramment utilisée en finance pour prendre en compte la variation des prix des actifs. Après cela, les données ont été normalisées dans une plage comprise entre [0, 1] à l'aide d'un scaler automatique pour faciliter l'entraînement et stabiliser les modèles. Enfin, les données ont été divisées en deux ensembles : 80% (3086) pour l'apprentissage et 20% (772) pour le test, permettant d'entraîner les modèles et d'évaluer leur performance sur des données nouvelles.



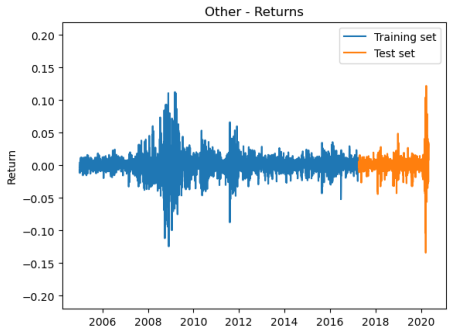


FIGURE 3.1 – Les rendements quotidiens de chaque secteur.

Ces graphiques montrent les mouvements des prix des actions dans les différents secteurs au fil du temps, illustrant ainsi les périodes de forte volatilité et de stabilité.

3.3 Statistiques descriptives et tests statistiques

Ces indications sont étudiées plus en détail dans le tableau ci-dessous qui fournit des statistiques descriptives pour l'ensemble de données de l'échantillon. La moyenne des cinq secteurs est proche de zéro, tandis que l'écart type donne un aperçu approximatif de leur volatilité inconditionnelle. Cela est également visible à travers l'écart min/max croissant des secteurs à mesure que l'écart type devient plus grand. L'asymétrie et l'aplatissement indiquent une non-normalité, ce qui est confirmé par la statistique de Jarque-Bera (Jarque-Bera (JB)) pour chaque secteur, rejetant l'hypothèse nulle de normalité au niveau de 1%. La statistique Ljung-Box Q rejette l'hypothèse nulle selon laquelle les données sont distribuées indépendamment jusqu'au $12^{ème}$ décalage pour tous les secteurs, indiquant la présence d'une autocorrélation. La statistique Ljung-Box Q^2, appliquée aux rendements carrés, indique la présence d'une hétéroscédasticité avec 12 décalages. Le test Augmented Dickey-Fuller (Augmented Dickey-Fuller (ADF)) rejette l'hypothèse nulle d'une racine unitaire dans tous les secteurs au niveau de 1%, indiquant que les rendements sont stationnaires.

	Statistiques descriptives						tests statistiques			
	Moyenne	L'écart type	Skewness	Kurtosis	Max	Min	JB	Q(12)	Q^2(12)	ADF
Consommation durable	0.00043	0.01037	-0.02916	8.82223	0.09880	-0.07270	9963.6	49.21	2937.2	-12.73
Santé	0.00043	0.01057	-0.04208	8.00669	0.11100	-0.06700	8206.1	55.56	2016.0	-14.64
Technologie	0.00044	0.01261	0.12667	9.41801	0.12470	-0.08510	11362.3	41.644	3029.5	-12.20
Industrie manufacturièr	0.00042	0.01366	-0.07920	12.54426	0.14520	-0.11860	20151.6	66.74	3577.2	-13.00
Autre	0.00031	0.01609	-0.05635	9.90497	0.11270	-0.12430	12562.4	60.43	3355.5	-11.01

TABLE 3.1 – Statistiques descriptives

3.4 Méthodologie d'estimation et de prédiction

3.4.1 Estimation avec le modèle EGARCH

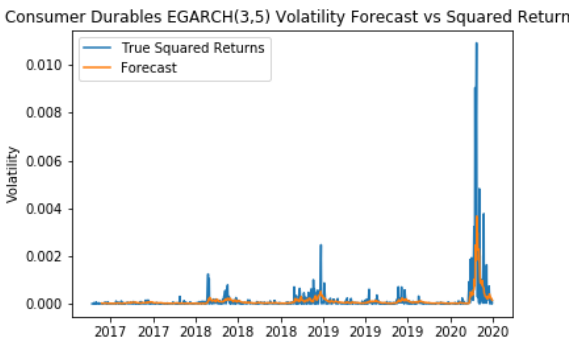
Pour chaque secteur, Curtis Nybo a estimé trois modèles GARCH différents (GARCH(p,q), GARCH(1,1) et EGARCH(p,q)) en utilisant les données de rendements quotidiens pour l'apprentissage. Le modèle le plus approprié a été sélectionné en fonction de la qualité de l'ajustement et du critère Akaike Information Criterion (AIC).

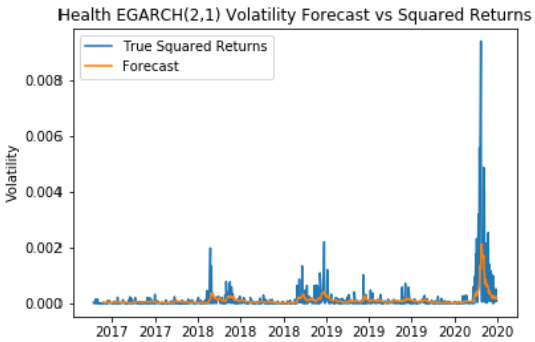
Le modèle EGARCH(p,q) s'est avéré être le plus adapté pour la prévision de la volatilité dans les cinq secteurs, car il capture bien les effets de la volatilité et gère les asymétries grâce au terme gamma. Les modèles EGARCH montrent des coefficients significatifs et des résidus de bruit blanc, assurant la stationnarité avec des termes β positifs et inférieurs à un. Les valeurs AIC plus basses pour le modèle EGARCH indiquent une meilleure capacité de prévision. En comparaison, les modèles GARCH(p,q) et GARCH(1,1) n'ont pas aussi bien capturé les effets de la volatilité et ont des valeurs AIC plus élevées.

	Consommation durable	Santé	Technologie	Industrie manufacturièr	Autre
MAE	0.0001291	0.0001479	0.0002108	0.0001788	0.0002231
MSE	0.0000003	0.0000002	0.0000006	0.0000006	0.0000008
RMSE	0.0005327	0.0004747	0.0007981	0.0007858	0.0009217

TABLE 3.2 – Performances du modèle EGARCH pour la prédiction de la volatilité par secteur.

Ce tableau montre les performances du modèle MAE en calculant des métriques entre les valeurs prédites et les valeurs réelles, notamment l'erreur absolue moyenne (MAE), l'erreur quadratique moyenne (MSE) et la racine carrée de l'erreur quadratique moyenne (RMSE). Ils fournissent une indication claire de la précision des prédictions. Des valeurs plus faibles indiquent une meilleure performance du modèle.





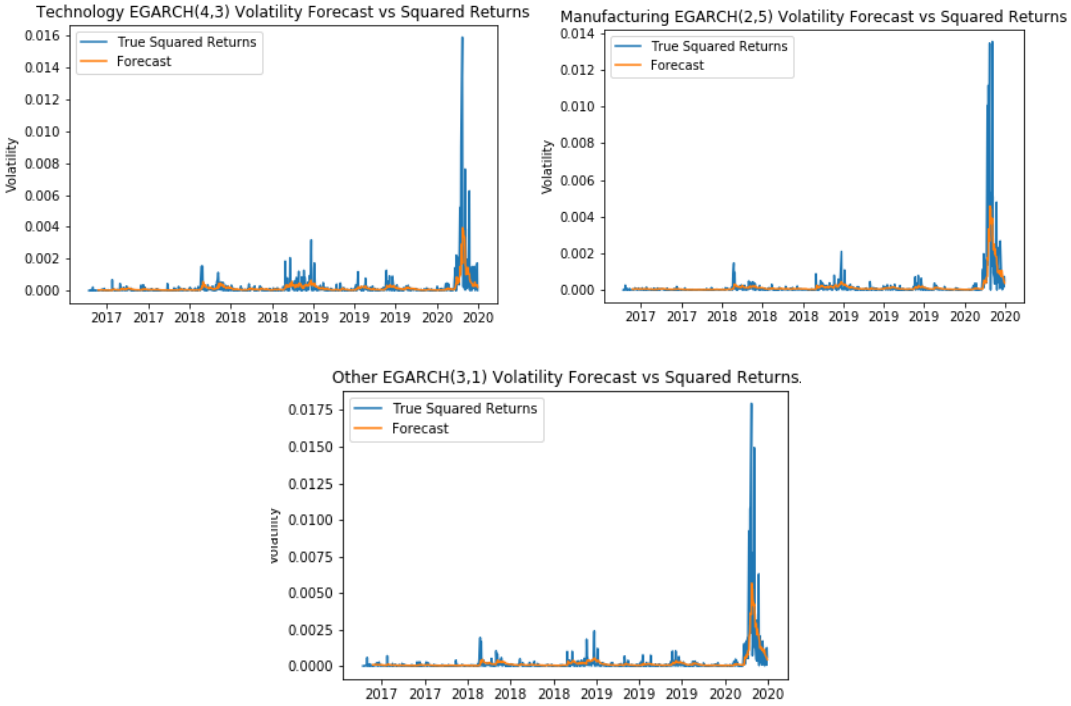


FIGURE 3.2 – Prédictions EGARCH comparées aux valeurs de volatilité réelles.

Ces graphiques comparent les fluctuations des prix des actions prédites par le modèle EGARCH avec les valeurs réelles observées, pour chaque secteur.

Maintenant que le modèle GARCH le plus approprié pour chaque secteur a été identifié, on peut calculer les variances inconditionnelles, car auparavant on ne pouvait utiliser que l'écart type comme indicateur de volatilité. Cela fournit une meilleure mesure globale de la volatilité puisqu'elle représente la volatilité moyenne à long terme.
La variance inconditionnelle des spécifications EGARCH de chaque secteur sont présentés ci-dessous :

	Secteur	**Variance inconditionnelle** (σ^2)	**Écart type incoditionnel** (σ)
Faible volatilité	Consommation durable	0.0000854	0.0092400
	Santé	0.0001077	0.0103792
Volatilité moyenne	Technologie	0.0001159	0.0107665
Forte volatilité	Industrie manufacturièr	0.0001325	0.0115114
	Autre	0.0001731	0.0131574

TABLE 3.3 – Variance inconditionnelle et écart-type pour différents secteurs.

3.4.2 Estimation avec ANN

Similaire aux spécifications du modèle GARCH, trois architectures de réseaux de neurones artificiels ont été entraînées sur les données d'entraînement. La même division train-test que

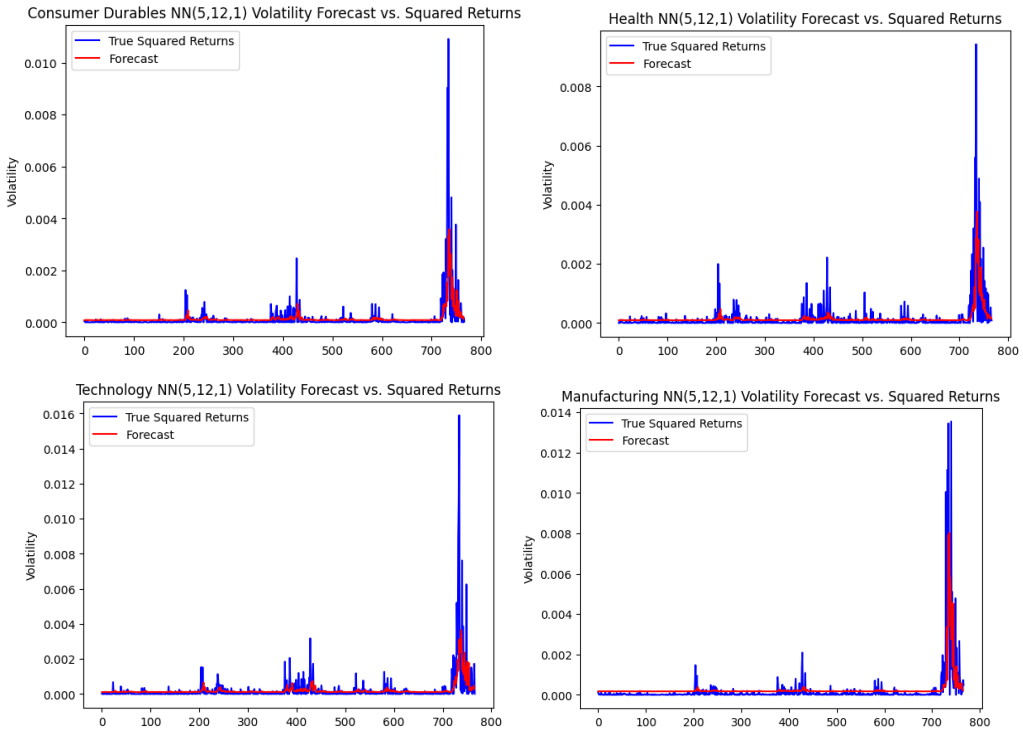
celle utilisée pour les modèles GARCH a été appliquée, sauf que 10% des données d'entraînement ont été utilisées comme ensemble de validation.

Cette étude teste la performance des trois architectures du modèle ANN 1, 12 et 50 neurones dans la couche cachée pour identifier celle qui performe le mieux, et pour trouver le nombre de neurone dans la couche cela se fait généralement par expérimentation. Dans chaque architecture, il y a toujours cinq neurones dans la couche d'entrée et un seul neurone dans la couche de sortie.

En termes de RMSE, l'erreur d'entraînement s'améliore avec 12 neurones, mais se dégrade avec 50 neurones, probablement à cause du surajustement. Ce modèle à 12 neurones est considéré comme le plus performant pour la prédiction de la volatilité dans chaque secteur. Voici le tableau ci-dessous qui présente les performances du modèle à 12 neurones, évaluées à l'aide des métriques MAE, MSE et RMSE.

	Consommation durable	Santé	Technologie	Industrie manufacturièr	Autre
MAE	0,0001595	0,0001752	0.0002454	0.0002636	0.0002915
MSE	0.0000003	0.0000002	0.0000006	0.0000007	0.0000009
RMSE	0,0005307	0,0004654	0.0008031	0.0008186	0.0009632

TABLE 3.4 – Performances du modèle ANN(5,12,1) pour la prédiction de la volatilité par secteur.



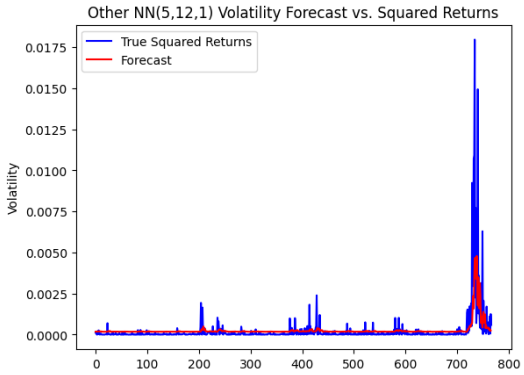


FIGURE 3.3 – Prédictions ANN(5,12,1) comparées aux valeurs de volatilité réelles.

Ces graphiques montrent comment le modèle ANN(5,12,1) prédit la volatilité des prix des actions par rapport aux valeurs réelles, pour chaque secteur.

3.4.3 Estimation avec les modèles de boosting

Pour les modèles de boosting, nous avons suivi une démarche rigoureuse pour optimiser leurs performances sur les différents secteurs. Après avoir divisé les données en ensembles d'entraînement, de validation et de test, nous avons ajusté chaque modèle en modifiant les paramètres à travers plusieurs tentatives jusqu'à obtenir une performance optimale de la prévision. Ces paramètres variaient en fonction des caractéristiques spécifiques des données de chaque secteur.

3.4.3.1 Estimation avec XGBoost

Pour XGBoost, nous avons déterminé que la profondeur des arbres doit être faible (environ 2) pour éviter le surajustement, avec un taux d'apprentissage modéré (environ 0,1 à 0,3) et un nombre d'estimations entre 100 et 200 pour obtenir des prédictions robustes. Ces paramètres permettent de capturer les relations importantes dans les données tout en évitant de surajuster les fluctuations aléatoires.

Prédiction et évaluation des performances de XGBoost
Après l'entraînement du modèle XGBoost sur l'ensemble d'entraînement, nous avons appliqué le modèle pour générer des prédictions sur les ensembles d'entraînement et de test. Les performances du modèle ont été évaluées en calculant des métriques d'évaluation. Ces mesures nous ont permis de vérifier la précision des prédictions du modèle XGBoost et d'évaluer sa capacité à capturer les tendances des données des différents secteurs.

	Consommation durable	Santé	Technologie	Industrie manufacturièr	Autre
MAE	0.0001474	0.0001622	0.0002402	0.0002084	0.0002306
MSE	0.0000003	0.0000002	0.0000006	0.0000006	0.0000008
RMSE	0.0005235	0.0004405	0.0007974	0.0007508	0.0008717

TABLE 3.5 – Performances du modèle XGBoost pour la prédiction de la volatilité par secteur.

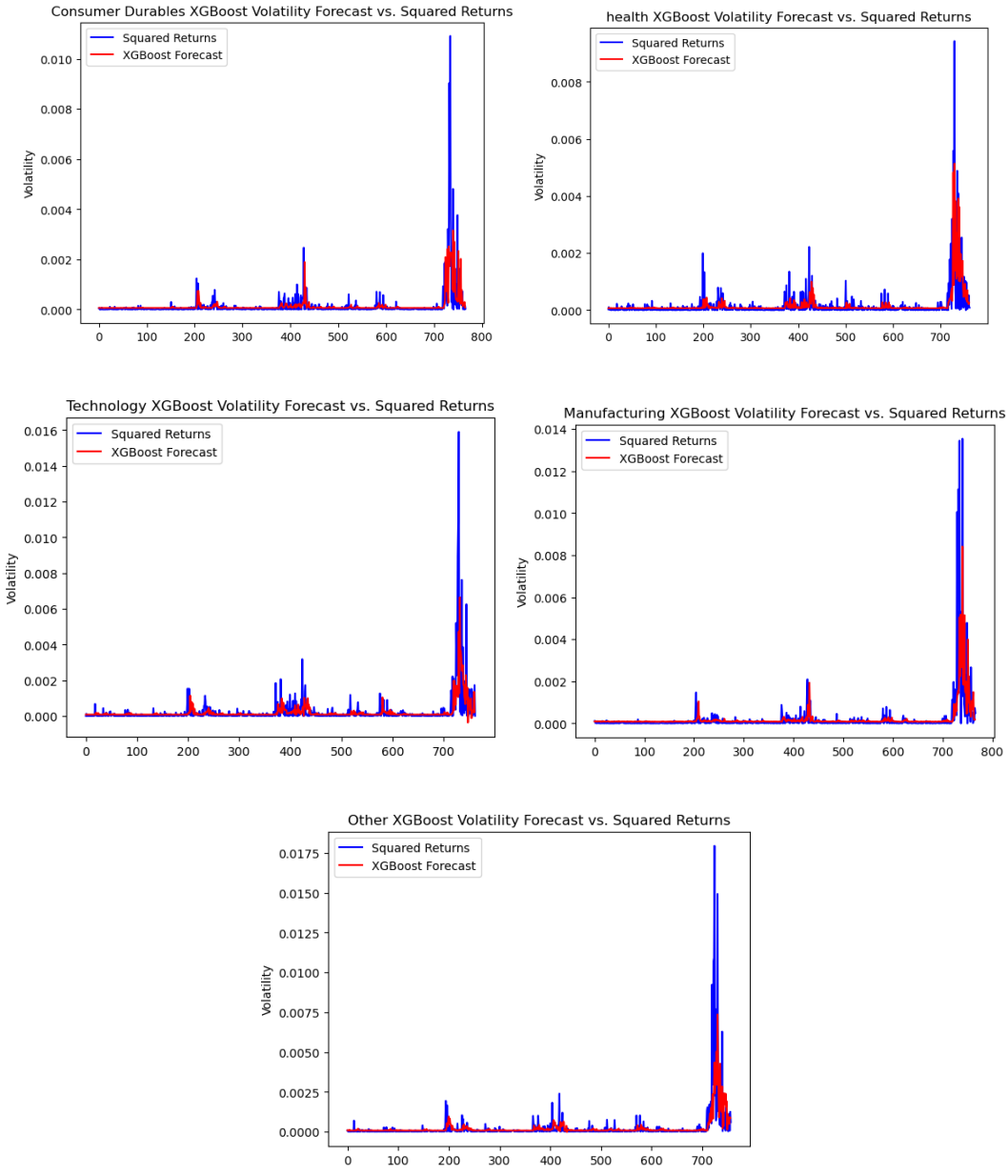


FIGURE 3.4 – Prédictions XGBoost comparées aux valeurs de volatilité réelles.

Ces graphiques illustrent les prédictions de la volatilité des prix des actions faites par le modèle XGBoost en comparaison avec les retours quadratiques réels, pour chaque secteur.

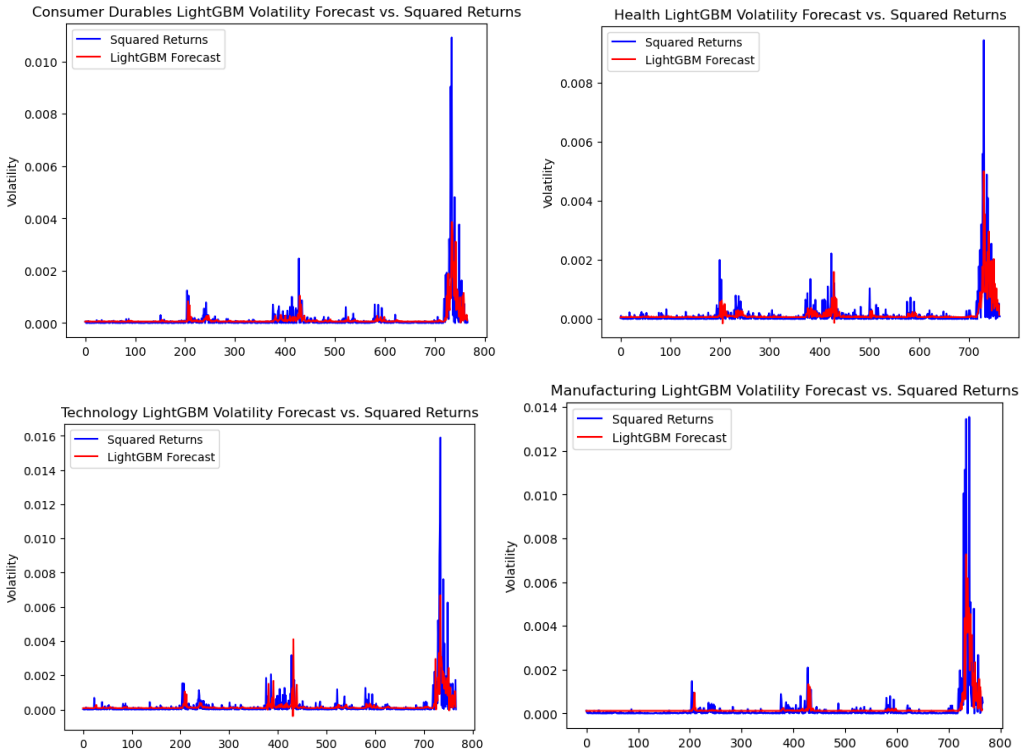
3.4.3.2 Estimation avec LightGBM

Avec LightGBM, nous avons optimisé des paramètres tels que la fraction de caractéristiques (autour de 0.8 à 1), la profondeur maximale des arbres (souvent fixée à 1 pour simplifier le modèle), et un nombre de feuille élevé (entre 100 à 500) pour garantir un bon équilibre entre complexité et généralisation. Une faible profondeur des arbres et un grand nombre d'estimations permettent d'améliorer la précision tout en contrôlant la complexité du modèle.

Prédiction et évaluation des performances de LightGBM
Pour le modèle LightGBM, nous avons suivi une procédure similaire à celle utilisée pour XGBoost. Les prédictions ont été transformées pour revenir à leur échelle originale et évaluées à l'aide des mêmes métriques (MAE, MSE, RMSE) afin de quantifier la précision et la robustesse du modèle.

	Consommation durable	Santé	Technologie	Industrie manufacturièr	Autre
MAE	0.0001430	0.0001513	0.0002313	0.0002177	0.0002491
MSE	0.0000003	0.0000002	0.0000006	0.0000006	0.0000008
RMSE	0.0005006	0.0004005	0.0007363	0.0007777	0.0009208

TABLE 3.6 – Performances du modèle LightGBM pour la prédiction de la volatilité par secteur.



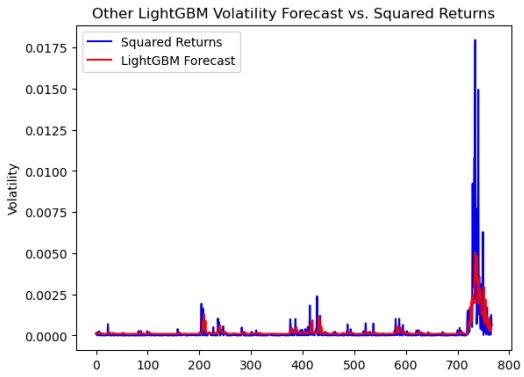


FIGURE 3.5 – Prédictions LightGBM comparées aux valeurs de volatilité réelles.

Ces graphiques mettent en évidence les prévisions de changements des prix des actions par le modèle LightGBM par rapport aux données réelles, pour chaque secteur.

3.4.3.3 Estimation avec CatBoost

Pour le modèle CatBoost, nous avons ajusté le nombre d'itérations (environ 100 à 500), le taux d'apprentissage (entre 0.05 à 0.9 pour une convergence rapide), et une faible profondeur des arbres (autour de 1) pour éviter le surajustement. Ces choix assurent une convergence rapide et réduisent le risque de surajustement, tout en permettant au modèle de capturer efficacement les relations dans les données.

Prédiction et évaluation des performances de CatBoost
De même, pour le modèle CatBoost, nous avons appliqué la même méthode que pour les modèles précédentes. Les prédictions ont été ramenées à leur échelle d'origine et les performances ont été mesurées en utilisant les métriques MAE, MSE et RMSE. Cela nous a permis de vérifier la capacité du modèle CatBoost à généraliser les données des différents secteurs et à fournir des prévisions précises de la volatilité.

	Consommation durable	Santé	Technologie	Industrie manufacturièr	Autre
MAE	0.0001453	0.0001600	0.0002306	0.0002161	0.0002521
MSE	0.0000003	0.0000002	0.0000006	0.0000006	0.0000008
RMSE	0.0005125	0.0004438	0.0007376	0.0007765	0.0009199

TABLE 3.7 – Performances du modèle CatBoost pour la prédiction de la volatilité par secteur.

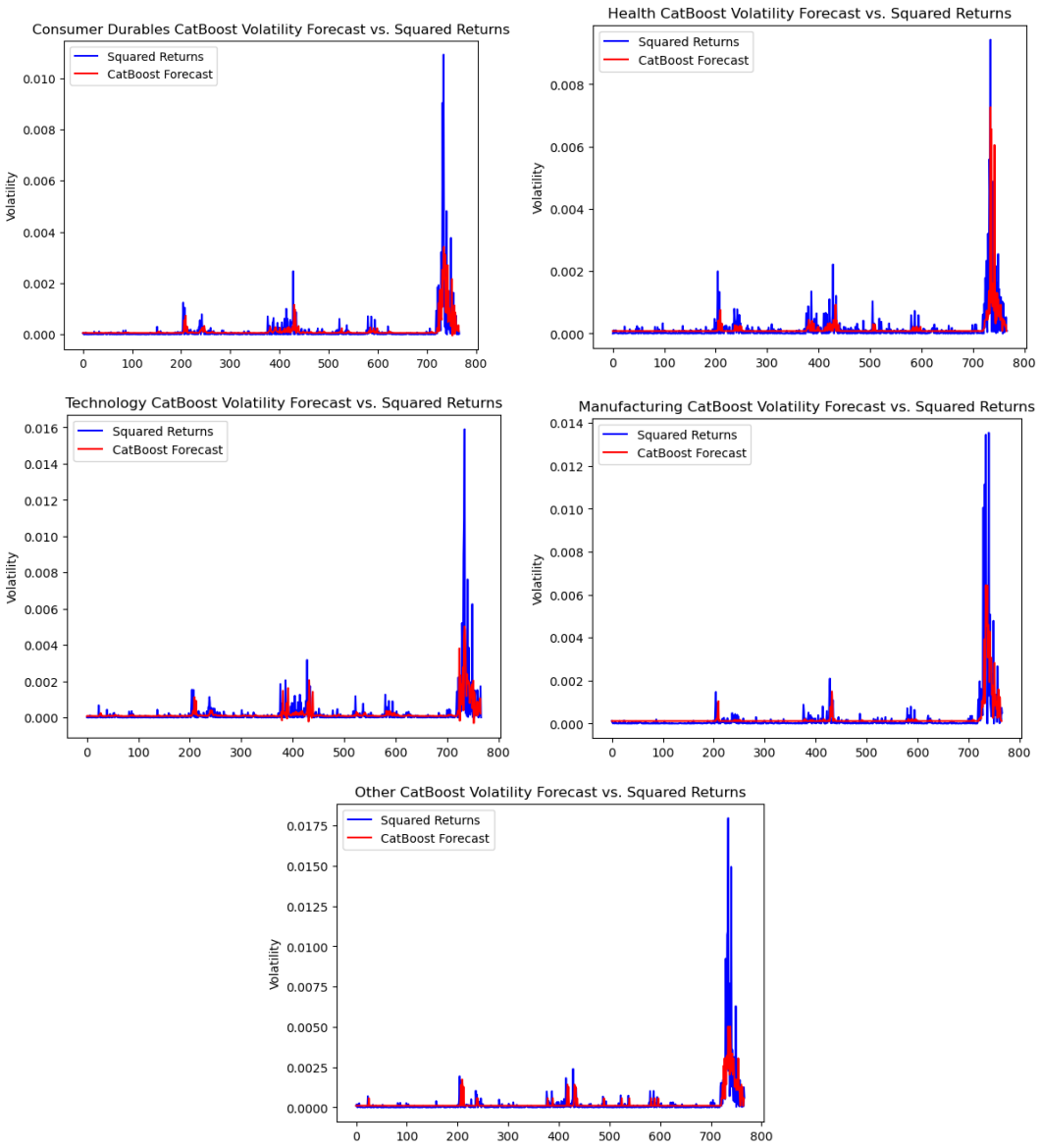


FIGURE 3.6 – Prédictions CatBoost comparées aux valeurs de volatilité réelles.

Ces graphiques démontrent comment le modèle CatBoost anticipe les fluctuations des prix des actions en comparaison avec les valeurs réelles, pour chaque secteur.

Les résultats obtenus montrent que ces modèles de boosting sont capables de capturer efficacement les tendances et les relations dans les données, offrant ainsi des prévisions fiables de la volatilité.

Les métriques d'évaluation telles que MAE, MSE et RMSE ont confirmé la capacité des modèles à généraliser les données, minimisant ainsi les erreurs de prédiction. Cette approche rigoureuse de l'optimisation des modèles de boosting nous a permis d'obtenir des outils performants pour l'analyse et la prévision de la volatilité dans différents secteurs.

La prochaine étape consistera à comparer ces modèles de boosting avec d'autres approches de modélisation pour évaluer leur efficacité relative et identifier les meilleures pratiques pour la prévision de la volatilité. Cette analyse comparative permettra de renforcer notre compréhension des dynamiques de volatilité et d'améliorer encore nos modèles prédictifs.

3.5 Analyse comparative des modèles de prédiction

Dans cette partie, nous comparons les performances des modèles de boosting (XGBoost), LightGBM et CatBoost avec celles des modèles traditionnels EGARCH et ANN pour la prédiction de la volatilité dans différents secteurs. Les métriques d'évaluation utilisées sont MAE, MSE et RMSE, cette dernière métrique déterminera quel modèle est le meilleure.

Consommation Durable	MAE	MSE	RMSE
EGARCH(3,5)	0.0001291	0.0000003	0.0005327
ANN(5,12,1)	0,0001595	0.0000003	0,0005307
XGBoost	0.0001474	0.0000003	0.0005235
LightGBM	0.0001430	0.0000003	0.0005006
CatBoost	0.0001453	0.0000003	0.0005125

TABLE 3.8 – Performances des modèles pour le secteur consommation durable.

Santé	MAE	MSE	RMSE
EGARCH(2,1)	0.0001479	0.0000002	0.0004747
ANN(5,12,1)	0,0001752	0.0000002	0,0004654
XGBoost	0.0001622	0.0000002	0.0004405
LightGBM	0.0001513	0.0000002	0.0004005
CatBoost	0.0001600	0.0000002	0.0004438

TABLE 3.9 – Performances des modèles pour le secteur santé.

Technologie	MAE	MSE	RMSE
EGARCH(4,3)	0.0002108	0.0000006	0.0007981
ANN(5,12,1)	0.0002454	0.0000006	0.0008031
XGBoost	0.0002402	0.0000006	0.0007974
LightGBM	0.0002313	0.0000006	0.0007363
CatBoost	0.0002306	0.0000006	0.0007376

TABLE 3.10 – Performances des modèles pour le secteur technologie.

Industrie Manufacturière	MAE	MSE	RMSE
EGARCH(2,5)	0.0001788	0.0000006	0.0007858
ANN(5,12,1)	0,0002636	0.0000007	0.0008186
XGBoost	0.0002084	0.0000006	0.0007508
LightGBM	0.0002177	0.0000006	0.0007777
CatBoost	0.0002161	0.0000006	0.0007765

TABLE 3.11 – Performances des modèles pour le secteur industrie manufacturière.

Autre	MAE	MSE	RMSE
EGARCH(3,1)	0.0002231	0.0000008	0.0009217
ANN(5, 12,1)	0.0002915	0.0000009	0.0009632
XGBoost	0.0002306	0.0000008	0.0008717
LightGBM	0.0002491	0.0000008	0.0009208
CatBoost	0.0002521	0.0000008	0.0009199

TABLE 3.12 – Performances des modèles pour le secteur autre.

L'analyse comparative des modèles de prédiction de la volatilité montre que les modèles de boosting, offrent des performances meilleur en termes de précision, flexibilité et capacité de généralisation. En utilisant la métrique RMSE comme critère principal pour déterminer le meilleur modèle, nous observons que ces modèles se distinguent dans plusieurs secteurs.

Le modèle LightGBM s'est avéré être le modèle le plus performant pour les secteurs de la consommation durable, de la santé et de la technologie, offrant un bon équilibre entre complexité et performance. Le modèle XGBoost, a montré une excellente performance dans les secteurs de l'industrie manufacturière et des autres secteurs. Quant au modèle CatBoost est souvent proches des meilleurs résultats.

Bien que les modèles GARCH et ANN soient utiles dans certains contextes, ils n'ont pas capturé aussi efficacement les dynamiques complexes des données de volatilité que les modèles de boosting.

3.6 Conclusion

Cette étude a comparé les performances de modèles de prédiction de la volatilité, incluant les modèles GARCH, ANN, XGBoost, LightGBM et CatBoost, à travers différents secteurs économiques. Les résultats montrent que les modèles de boosting, surpassent généralement les modèles traditionnels en termes de précision et de généralisation. LightGBM s'est avéré le plus performant dans les secteurs de la consommation durable, de la santé et de la technologie, tandis que le modèle XGBoost a excellé dans l'industrie manufacturière et d'autres secteurs. Cependant, les modèles de boosting sont recommandés pour des prévisions de volatilité plus précises et robustes.

CONCLUSION GÉNÉRALE

L'analyse comparative menée dans ce mémoire a révélé que les modèles de boosting modernes, notamment XGBoost et LightGBM, surpassent les modèles traditionnels (GARCH et ANN) pour la prédiction de la volatilité des marchés financiers. Ces modèles d'apprentissage automatique se sont avérés plus précis et robustes face à la complexité des données financières, démontrant leur capacité à capturer des patterns subtils et à s'adapter aux fluctuations dynamiques des marchés.

Nos résultats, s'appuyant sur des données de cinq secteurs économiques, confirment l'intérêt croissant des techniques d'apprentissage automatique pour la prévision de la volatilité. Ils suggèrent que ces modèles offrent un potentiel significatif pour améliorer la prise de décision en gestion des risques et en planification financière.

Pour les recherches futures, il serait pertinent d'explorer l'optimisation des modèles de boosting et de tester leur performance dans d'autres contextes économiques. L'intégration de nouvelles techniques d'apprentissage automatique, telles que l'apprentissage profond, pourrait également contribuer à améliorer la précision des prévisions de volatilité et offrir de nouvelles perspectives pour la gestion des risques financiers.

BIBLIOGRAPHIE

[1] Al Daoud, E., Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset, International Journal of Computer and Information Engineering, 13(1), 6-10, (2019).

[2] Amine, M., Orientation et réparation en garages partenaires en assurance automobile, Mémoire de Maîtrise, Ecole Nationale de la statistique et de l'administration economique (ENSAE) 5, avenue Henry Le Chatelier- 91120 Palaiseau, France, (2021).

[3] Augustyniak, M., Estimation du modèle GARCH à changement de régimes et son utilité pour quantifier le risque de modèle dans les applications financières en actuariat, Philosophiæ Doctor (Ph. D.) en Statistique, Université de Montréal, (2014).

[4] Bouras, E. H., Besoin en eau et rendements des céréales en Méditerranée du Sud : observation, prévision saisonnière et impact du changement climatique, Doctoral dissertation, Université Paul Sabatier-Toulouse III ; Université Cadi Ayyad (Marrakech, Maroc), (2021).

[5] Breiman, L., Friedman, J., Stone, C., and Olshen, R., Classification and regression trees, Routledge, (1984).

[6] Candillier, L., Contextualisation, visualisation et évaluation en apprentissage non supervisé, Doctoral dissertation, Université Charles de Gaulle-Lille III, (2006).

[7] Cerisier, V., Application de méthodes de machine learning dans le calcul de la solvabilité infraannuelle, Université de Paris-Dauphine pour l'obtention du Certificat d'Actuaire de Paris-Dauphine, (2021).

[8] Chamekh, A., Optimisation des procédés de mise en forme par les réseaux de neurones artificiels, Doctoral dissertation, Université d'Angers, (2008).

[9] Chen, T., Guestrin, C., Xgboost : A scalable tree boosting system, In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794), (2016).

[10] Djabrane, Y., Séries temporelles et test d'adéquation pour un modèle GARCH(1, 1), Doctoral dissertation, Université Mohamed Khider Biskra, (2005).

[11] Dutertre-Laduree, L., Gardahaut, A., and Tsang, G., Tarification des contrats d'assurance automobile, Universite de Rennes 1, (2020).

[12] Février, M. C., Prévision des défaillances des entreprises avec la bibliothèque de Machine Learning XGBoost, Mastère 2 in Artificial Intelligence and Management délivré par l'IA School, (2021).

[13] Freund, Y., and Schapire, R. E., Experiments with a new boosting algorithm, In icml (Vol. 96, pp. 148-156), (1996).

[14] Friedman, J. H., Greedy function approximation : a gradient boosting machine, Annals of statistics, 1189-1232, (2001).

[15] Gauville, R., Projection du ratio de solvabilité : des méthodes de machine learning pour contourner les contraintes opérationnelles de la méthode des SdS, Diplôme d'Actuaire EURIA et de l'admission à l'Institut des Actuaires, (2017).

[16] Genuer, R., Forêts aléatoires : aspects théoriques, sélection de variables et applications, Doctoral dissertation, Université Paris Sud-Paris XI, (2010).

[17] Hancock, J. T., and Khoshgoftaar, T. M., CatBoost for big data : an interdisciplinary review, Journal of big data, 7(1), 94, (2020).

[18] Hemsas, O., Prédiction dans les séries chronologiques, Doctoral dissertation, UMMTO, (2014).

[19] Hurlin, C., and Pérignon, C., Machine learning et modèles IRB : avantages, risques et préconisations, Doctoral dissertation, Institut Louis Bachelier, (2023).

[20] Jerbi, J., Evaluation des options et gestion des risques financiers par les réseaux de neurones et par les modèles à volatilité stochastique, Doctoral dissertation, Université Panthéon-Sorbonne-Paris I, (2006).

[21] Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., and Rehman, M. U., A model combining convolutional neural network and LightGBM algorithm for ultra-short-term wind power forecasting, Ieee Access, 7, 28309-28318, (2019).

[22] Kamran, M., A state of the art catboost-based T-distributed stochastic neighbor embedding technique to predict back-break at dewan cement limestone quarry, Journal of Mining and Environment, 12(3), 679-691, (2021).

[23] Koye, G. K., Comparaison des méthodes classiques et alternatives avec le machine learning pour la construction d'une table de mortalité d'experience best estimate, Ecole Nationale de la statistique et de l'administration economique (ENSAE) 5, avenue Henry Le Chatelier-91120 Palaiseau, France, (2019).

[24] Kozhemyak, A., Modélisation de séries financières à l'aide de processus invariants d'échelle. Application à la prédiction du risque, Doctoral dissertation, Ecole Polytechnique X, (2006).

[25] Lahcen, D. A., Amghar, N. E., and Oukassi, M., La défaillance des entreprises : Regard comparatif sur les diverses approches et lecture critique des divers modèles prédictifs, International Journal of Advanced Research in Innovation, Management and Social Sciences, 6(1), (2023).

[26] Lecours, M. A., Modélisation de la structure de la variance et du coefficient d'asymétrie d'options sur indice à l'aide d'un modèle Garch, Doctoral dissertation, HEC Montréal, (2020).

[27] Mahamat, H. S., Estimation de la volatilité des données financières à haute fréquence : une approche par le Modèle Score-GARCH, Doctoral dissertation, Université Montpellier, (2017).

[28] Mattei, P. A., and Villata, S., Introduction à l'intelligence artificielle et aux modèles génératifs, Université Côte d'Azur, Inria, (2022).

[29] Nybo, C., Sector volatility prediction performance using GARCH models and artificial neural networks, arXiv preprint arXiv :2110.09489, (2021).

[30] Ottou, P., Méthodes d'apprentissage automatique appliquées au provisionnement ligne à ligne en assurance non-vie, Université Paris Dauphin, (2017).

[31] Paquet, P., L'utilisation des réseaux de neurones artificiels en finance, Document de recherche n° 1997-1, (1997).

[32] Parizeau, M., Réseaux de neurones, GIF-21140 et GIF-64326, Université laval, (2004).

[33] Ponthier, L., Application des approches de modelisation et de machine learning a l'individualisation des doses d'anti-infectieux en pediatrie, Doctoral dissertation, Université de Limoges, (2023)

[34] Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A., CatBoost : unbiased boosting with categorical features, Advances in neural information processing systems, 31, (2018).

[35] Rouane, S., El houda Rouane, N., and Yahia, M. S. B., Indentification par réseaux de neurone, Université de M'Sila, (2019).

[36] Rufo, D. D., Debelee, T. G., Ibenthal, A., and Negera, W. G., Diagnosis of diabetes mellitus using gradient boosting machine (LightGBM), Diagnostics, 11(9), 1714, (2021).

[37] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., Learning representations by back-propagating errors, nature, 323(6088), 533-536, (1986).

[38] Saidane, M., Modèles à facteurs conditionnellement hétéroscédastiques et à structure markovienne cachée pour les séries financières, Doctoral dissertation, Université Montpellier II-Sciences et Techniques du Languedoc, (2006).

[39] Servel, M., Une approche individuelle du provisionnement des sinistres corporels automobiles, Institut de science financière et d'assurances, (2020).

[40] Sy, A., La volatilité stochastique des marchés financiers : une application aux modèles d'évaluation d'instruments optionnels en temps continu, Doctoral dissertation, Université de droit, d'économie et des sciences-Aix-Marseille III, (2003).

[41] Wang, Y., and Wang, T., Application of improved LightGBM model in blood glucose prediction, Applied Sciences, 10(9), 3227, (2020).

[42] Wira, P., Réseaux de neurones artificiels : architectures et applications, Cours en ligne, Université de Haute-Alsace, (2009).

[43] Yahia, D., Benatia, F., and Touba, S., Les lois α-stables comme modèle pour les séries financières, Mémoire de fin d'étude, Université Mohamed Khider Biskra, (2012).

Résumé

Dans ce mémoire, nous abordons la problématique cruciale de la prédiction de la volatilité des marchés financiers, essentielle pour les investisseurs et les gestionnaires de risques. Nous comparons la performance des modèles traditionnels tels que GARCH, avec des modèles d'apprentissage automatique modernes, notamment le réseau de neurones artificiels (ANN), XGBoost, LightGBM et CatBoost. Notre travail repose sur l'analyse de données provenant de cinq secteurs industriels distincts, et nous évaluons l'efficacité de chaque modèle en utilisant des métriques fiables telles que MAE, MSE et RMSE. En identifiant le modèle le plus précis pour chaque secteur, nous proposons des recommandations concrètes pour améliorer la gestion des risques financiers et optimiser les stratégies d'investissement. Notre étude met en lumière l'importance des approches innovantes pour prédire les fluctuations du marché et offrir des outils fiables aux décideurs financiers.

Mots clés

Volatilité, Marchés financiers, Prédiction, Modèles GARCH, Apprentissage automatique, Réseaux de neurones artificiels (ANN), XGBoost, LightGBM, CatBoost, Les métriques d'évaluation (RMSE, MAE, MSE).

Abstract

In this thesis, we address the crucial problem of predicting financial market volatility, which is essential for investors and risk managers. We compare the performance of traditional models such as GARCH with modern machine learning models, including Artificial Neural Networks (ANN), XGBoost, LightGBM, and CatBoost. Our work is based on data collected from five different industrial sectors, and we evaluate the effectiveness of each model using reliable metrics such as MAE, MSE, and RMSE. By identifying the most accurate model for each sector, we provide practical recommendations to improve financial risk management and optimize investment strategies. Our study highlights the importance of innovative approaches to predict market fluctuations and provides reliable tools for financial decision-makers.

Keywords

Volatility, Financial markets, Prediction, GARCH models, Machine learning, Artificial Neural Networks (ANN), XGBoost, LightGBM, CatBoost, evaluation metrics (RMSE, MAE, MSE).