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

Université A. Mira de Béjaia
Faculté des Sciences Exactes
Département de Recherche Opérationnelle

Mémoire de fin de cycle
en vue de l'obtention du diplôme de master
en Mathématiques appliquées

Option : Science des données et aide à la décision

Thème :

Apprentissage automatique pour la prévision des ventes Cas d'étude : Cevital SPA

Présenté par :

M^lle^ CHIKH Yasmine
M. OUGUERGOUZ Mehdi

Soutenu devant le jury composé de :

ASLI Larbi	Université de Béjaia	Président
BACHI Katia	Université de Béjaia	Examinatrice
BOUROUINA Massilva	Université de Béjaia	Examinatrice
DJABRI Rabah	Université de Béjaia	Encadrant

Année universitaire : 2023/2024

Remerciments

Avant tout, nous exprimons notre profonde gratitude à Allah, le Tout-Puissant, de nous avoir accordé la force et la patience nécessaires pour pour achever ce travail jusqu'à son terme. Sans sa bénédiction et son soutien constant, nous n'aurions pas pu surmonter les nombreux défis rencontrés tout au long de cette aventure académique.

Nous tenons à exprimer notre reconnaissance à notre encadrant, Dr. Rabah DJABRI, pour son accompagnement et ses conseils avisés. Sa patience et sa guidance ont été essentiels à la réussite de ce projet.

Nous adressons également nos remerciements les plus sincères à tous les professeurs de l'Université, en particulier à Dr. Larbi ASLI, Dr. Zoubeyr FARAH, Dr. Nadim RAGAB et à Dr. Mohamed MOHAMMEDI pour leur soutien infaillible et leurs précieux conseils qui ont été déterminants pour notre progression. Leur expertise, leur disponibilité et leurs encouragements constants ont ont joué un rôle majeur dans développement académique et professionnel. Chacun d'eux a contribué de manière significative en nous apportant les connaissances et les compétences nécessaires pour mener à bien ce projet.

Nous souhaitons exprimer notre gratitude à toute l'équipe de Cevital SPA, et en particulier à M. Athmane CHIKH, notre encadrant sur le terrain, qui nous a guidés et soutenus tout au long de notre projet au sein de l'entreprise.

Nous remercions également les membres du jury pour l'honneur qu'ils nous font en évaluant notre travail, ainsi que pour leurs critiques constructives qui seront précieuses pour l'amélioration de notre projet. Nous sommes reconnaissants pour le temps et l'attention qu'ils consacrent à l'évaluation de notre travail.

Enfin, nous adressons nos remerciements à toutes les personnes qui nous ont soutenus, de près ou de loin, tout au long de la réalisation de ce projet. Vos encouragements, votre aide et vos mots de soutien ont été des sources de motivation et de réconfort.

Merci infiniment à tous.

Dédicaces

La vie est un chemin jalonné de présences précieuses et de soutiens inestimables. Je dédie ce travail à toutes ces personnes sans les quelles ma vie ne serait pas la même.

À **mes très chers parents**, qui ont consacré leur vie à ma réussite. Votre amour sans bornes, vos sacrifices incessants et votre dévouement absolu ont été mes plus grands atouts. Vous m'avez guidé avec une sagesse et un soutien constants. Je vous dois tout et vous exprime ma gratitude infinie. Merci de m'avoir toujours épaulé, de continuer à le faire sans faillir et de me soutenir avec constance dans chaque décision et chaque étape de ma vie. Je vous en suis éternellement reconnaissante. Je vous dédie ce travail en témoignage de mon profond amour.

À **mes chers frères** ***(Wassim et Yanis)***, pour votre force, votre motivation et votre réconfort qui ont été indispensables pour surmonter chaque obstacle. Je vous exprime mes plus grands sentiments de fraternité et d'affection, que vos vies soient remplies de bonheur, de succès et de la même détermination que vous m'avez inspirée.

À **mes amis**, à qui je souhaite un avenir radieux.

À **moi-même**, en hommage à ma persévérance et ma diligence, mes efforts constants et ma détermination depuis mon jeune âge ont été essentiels pour mon succès.

À **la mémoire** de mes chers grands-parents défunts, qui auraient été immensément fiers de mes accomplissements.

Une pensée pour ma carrière artistique et la sortie récente de mon single ***El Feniw***, et pour l'achèvement de mon cycle de Master, deux accomplissements qui symbolisent la finalité de deux chapitres importants de ma vie et dont je suis extrêmement fière.

DIEU MERCI.

Yasmine.

Dédicaces

À la mémoire de mon grand-père, **Daoud Mohamed**, Allah yerahamou, dont la sagesse et l'amour continuent de m'inspirer chaque jour. Il était fier de voir ses petits-enfants réussir dans leurs études, son esprit bienveillant reste une source constante de motivation pour nous tous.

À mes parents, pour leur amour infini, leur soutien inconditionnel et les nombreux sacrifices qu'ils ont consentis pour m'offrir les meilleures opportunités d'éducation. Votre force et votre dévouement sont les fondations de tout ce que j'ai entrepris. Sans vous, rien de tout cela n'aurait été possible.

À mon cher frère **Khaled** et à ma chère sœur **Imene**, pour leur soutien indéfectible tout au long de mes études. Vous avez été des piliers essentiels, m'encourageant dans les moments difficiles et partageant ma joie dans les réussites.

À **Dr. ZENNACHE Lydia** et à mes amis **CHEBIHI Salma**, **CHIBANE Zebida** et **Redjal Zineddine**, pour leur amitié sincère et leurs encouragements constants. Vous avez rendu chaque obstacle plus surmontable et chaque moment en un souvenir précieux. Votre présence à mes côtés a été une bénédiction.

Et à tous ceux qui nous aiment et que nous aimons, de près ou de loin, et qui ont contribué à la réalisation de ce mémoire de fin d'études. Vos gestes, vos paroles et votre soutien ont été inestimables tout au long de ce parcours.

 # FREE PALESTINE.

Mehdi.

Table des matières

Remerciements I
Dédicaces II
Dédicaces III
Liste des figures VIII
Liste des tables IX
Liste des abréviations et notations IX

Introduction générale **1**

1 Prérequis théoriques et présentation de l'organisme d'accueil **3**
Introduction 3
1.1 Présentation de Cevital 4
1.2 Situation géographique 4
1.3 Organigramme de Cevital 5
1.4 Missions du département cible : Reporting et analyse des ventes 6
1.5 Défis et contraintes 7
1.6 Problématique 7
1.7 Solution proposée 8
1.8 Principes des prévision des ventes 8
1.8.1 Ventes 8
1.8.2 Prévision 8
1.8.3 Prévision des ventes 9
1.8.4 Importance et objectifs 9
1.8.5 Méthodes de la prévision des ventes 9
1.9 Fondements des séries chronologiques 10
1.10 Modèles d'analyse des séries chronologiques 13
1.11 Modèles de prévision et de modélisation 14
1.11.1 Modèle auto-régressif AR(p) 15
1.11.2 Moyennes mobiles MA(q) 15
1.11.3 Modèles ARMA(p,q) et ARIMA(p,d,q) 15
1.11.4 Modèle SARIMA $((p,d,q)\times(P,D,Q))_s$ 16
1.11.5 Box-Jenkins 17
Conclusion 18

2 Apprentissage automatique **19**
Introduction 19
2.1 Intelligence artificielle 20

2.2 Apprentissage automatique . . . 20
2.2.1 Phases de l'apprentissage automatique . . . 23
2.2.2 Limites de l'apprentissage automatique . . . 24
2.3 Types d'apprentissage automatique . . . 24
2.3.1 Apprentissage supervisé . . . 25
2.3.2 Apprentissage non supervisé . . . 26
2.3.3 Apprentissage semi-supervisé . . . 27
2.3.4 Apprentissage par renforcement . . . 27
2.4 Algorithmes de l'apprentissage automatique . . . 27
2.4.1 Régression linéaire . . . 28
2.4.2 K-nearest neighbors (KNN) . . . 28
2.4.3 Arbre de décision . . . 30
2.4.4 Méthodes d'ensemble séquentielles (boosting) . . . 32
2.4.5 Méthodes d'ensemble parallèles (bagging) . . . 37
2.4.6 Stacking . . . 39
2.5 Apprentissage profond . . . 40
2.6 Optimisation et validation des modèles . . . 40
2.7 Métriques d'évaluation des modèles . . . 41
2.8 Travaux connexes . . . 42
2.8.1 Apprentissage automatique supervisé . . . 42
2.8.2 Apprentissage profond . . . 44
Conclusion . . . 48

3 Méthodologie de l'étude **49**
Introduction . . . 49
3.1 Schéma de la solution proposée . . . 49
3.2 Environnement de développement . . . 51
3.2.1 Langage de programmation . . . 51
3.2.2 Bibliothèques et frameworks pour la data science avec Python . . . 51
3.2.3 Environnement de développement intégré IDE . . . 53
3.2.4 Outil de visualisation et de reporting . . . 53
3.3 Cas d'étude . . . 53
3.3.1 Collecte et chargement des données . . . 53
3.3.2 Nettoyage des données . . . 54
3.3.3 Prétraitement des données . . . 54
3.3.4 Analyse exploratoire des données . . . 55
3.3.5 Décomposition saisonnière . . . 58
3.3.6 Traitement des données . . . 59
3.4 Modèle classique SARIMA . . . 60
3.5 Modèles d'apprentissage automatique . . . 62
3.5.1 Forêt aléatoire . . . 62
3.5.2 XGBoost . . . 63
3.5.3 CatBoost . . . 64
3.5.4 Hybridation SARIMA-CatBoost . . . 64
Conclusion . . . 66

4 Résultats et discussions **67**
Introduction 67
4.1 Présentation des résultats 67
4.1.1 Prévisions avec le modèle SARIMA 68
4.1.2 Prévisions avec le modèle forêt aléatoire 69
4.1.3 Prévisions avec le modèle XGBoost 69
4.1.4 Prévisions avec le modèle CatBoost 70
4.1.5 Prévisions avec le modèle hybride SARIMA-CatBoost 71
4.1.6 Évaluation comparative des modèles 73
4.2 Évaluation de divers algorithmes sur plusieurs séries chronologiques 75
4.3 Tableau de bord 76
Conclusion 78

Conclusion générale **79**

Bibliographie **82**

Résumé **83**

Liste des figures

1.1 Logo de l'entreprise Cevital et ses différentes marques de produits 4
1.2 Situation géographique du complexe Cevital 5
1.3 Organigramme de Cevital 6
1.4 Composantes d'une série chronologique 13
1.5 Représentation des modèles graphiques 14
1.6 Étapes de la méthodologie Box-Jenkins 17

2.1 Panorama de l'intelligence artificielle 20
2.2 Descente de gradient 23
2.3 Phases de l'apprentissage automatique 24
2.4 Types d'apprentissage automatique 25
2.5 Exemple des tâches de l'apprentissage supervisé 26
2.6 Exemple d'apprentissage automatique non supervisé 26
2.7 Apprentissage par renforcement 27
2.8 Exemple d'un KNN 30
2.9 Exemple d'un arbre de décision 31
2.10 Entraînement d'AdaBoost 33
2.11 Processus d'assemblage 40

3.1 Schéma de la solution proposée 50
3.2 Logo de Python 51
3.3 Logo de Jupyterlab 53
3.4 Chargement des données 54
3.5 Nettoyage des données 54
3.6 Statistiques descriptives 55
3.7 Ventes totales par sous-famille 56
3.8 Ventes totales par canal 56
3.9 Ventes totales par région 57
3.10 Répartition des ventes par sous-famille et canal 57
3.11 Répartition des ventes par sous-famille et région 58
3.12 Évolution des ventes 58
3.13 Décomposition saisonnière 59
3.14 Fonctions ACF et PACF 61

4.1 Résultats avec SARIMA 68
4.2 Résultats avec forêt aléatoire 69
4.3 Résultats avec XGBoost 70

4.4 Résultats avec CatBoost 71
4.5 Résultats avec le modèle hybride SARIMA-CatBoost 72
4.6 Comparaison de la métrique R^2 74
4.7 Comparaison des métriques RMSE et MAE 74
4.8 Comparaison des prévisions des différents algorithmes 74
4.9 Tableau comparatif des performances des modèles de prévision 75
4.10 Feuille n°1 du tableau de bord 76
4.11 Feuille n°2 du tableau de bord 77

Liste des tableaux

2.1 Tableau des algorithmes d'apprentissage automatique par type d'apprentissage et tâches 28
2.2 Tableau récapitulatif des travaux connexes 47

4.1 Tableau comparatif des performances des modèles de prévision 73

Liste des abréviations et notations

ACF	Autocorrelation function
ADF	Augmented Dickey-Fuller test
AIC	Akaike Information Criterion
AR	Autoregressive model
ARIMA	Autoregressive integrated moving average
ARMA	Autoregressive moving average
BI	Business Intelligence
CART	Classification and regression trees
DL	Deep learning
DT	Decision tree
GB	Gradient boost
GBT	Gradient boosted trees
GLM	Modèle linéaire généralisé
IA	Intelligence artificielle
IBM	International Business Machines Corporation
KNN	K−nearest neighbors
KPI	Key performance indicator
LSTM	Long short-term memory
MA	Moving-average model
MAE	Mean absolute error
ML	Machine learning
MSE	Mean squared error
PACF	Partial autocorrelation function
RF	Random forest
RG	Régression logistique
RMSE	Root mean squared error
RNN	Réseaux de neurones récurrents
SARIMA	Seasonal autoregressive integrated moving average
SKU	Stock Keeping Unit
SPA	Société par actions
SVM	Support vector machine

Introduction générale

Prévoir, ce n'est pas prédire l'avenir mais le rendre possible. - Antoine de Saint-Exupéry

Dans un monde où les données sont essentielles pour les décisions stratégiques, l'intégration de l'apprentissage automatique révolutionne la prévision des ventes, permettant une analyse avancée des données et une anticipation proactive des tendances du marché. Les entreprises d'aujourd'hui gèrent d'énormes répertoires de données, dont le volume est attendu pour croître de manière exponentielle, elles doivent constamment s'adapter aux fluctuations du marché et aux attentes changeantes des consommateurs pour rester compétitives. Cevital SPA, leader du secteur agroalimentaire en Algérie, se trouve au cœur de ces défis. Avec une présence diversifiée dans des secteurs tels que l'agroalimentaire, l'industrie et les services, Cevital cherche à optimiser ses chaînes d'approvisionnement et à maximiser ses profits grâce à des outils de prévision des ventes précis et robustes.

La prévision des ventes joue un rôle crucial dans la planification stratégique, la gestion des stocks et la satisfaction des clients. Cependant, les méthodes classiques de prévision, bien qu'efficaces, montrent des limites en termes de précision et de réactivité face aux changements rapides et imprévisibles du marché. C'est ici que l'apprentissage automatique intervient, offrant des solutions plus flexibles et capables de traiter des volumes de données considérables, tout en capturant des relations complexes et non linéaires au sein de ces données.

La motivation principale de ce mémoire réside dans la nécessité de répondre aux défis actuels de prévision des ventes auxquels fait face Cevital SPA. Avec la complexité croissante du marché et la nécessité d'optimiser les chaînes d'approvisionnement, les méthodes classiques de prévision montrent leurs limites en termes de précision, de réactivité et de flexibilité. Ces méthodes, bien qu'ayant servi efficacement par le passé, ne sont plus adaptées à la vitesse et à la complexité des dynamiques du marché actuel.

L'apprentissage automatique permet de surmonter les limites des méthodes classiques de prévision des ventes en traitant de grandes quantités de données et en produisant des prévisions plus précises. Pour exploiter pleinement ces capacités, ce mémoire propose l'intégration de techniques avancées d'apprentissage automatique avec les méthodes classiques de prévision. Nous allons nous concentrer sur des sous-familles d'un produit précis de Cevital SPA pour démontrer l'application pratique de ces techniques. Nous comparerons plusieurs algorithmes d'apprentissage automatique notamment SARIMA, forêts aléatoires, CatBoost, XGBoost, ainsi que des modèles hybrides (SARIMA-CatBoost, SARIMA-XGBoost).

Ce mémoire est organisé en quatre chapitres agencés de la manière suivante :

- Le premier chapitre se portera sur la présentation de notre organisme d'accueil et introduira les concepts fondamentaux des séries chronologiques, les techniques de prévision traditionnelles, et présente un cadre théorique pour comprendre l'importance de la prévi-

sion des ventes dans un contexte commercial moderne.

- Le deuxième chapitre se concentrera sur l'apprentissage automatique, détaillant les principes et les algorithmes clés, et explorant leurs applications spécifiques à la modélisation des séries chronologiques.
- Le troisième chapitre détaillera la mise en œuvre des différents modèles prédictifs, en décrivant les processus de traitement et de modélisation des données, les outils logiciels utilisés, ainsi que la création et l'entraînement des modèles traditionnels et ceux d'apprentissage automatique.
- Enfin, le quatrième chapitre exposera les résultats expérimentaux et les analyses comparatives des performances des différents modèles, illustrant les avantages et les améliorations apportés par l'approche hybride dans le contexte de la prévision des ventes. De plus, ce chapitre proposera une visualisation claire et intuitive de ces résultats à travers l'utilisation de tableaux de bord interactifs.
- En conclusion, ce mémoire ambitionne de démontrer comment l'intégration des techniques classiques et modernes de modélisation des séries chronologiques peut révolutionner la prévision des ventes, offrant ainsi aux entreprises un outil puissant pour naviguer dans un environnement commercial de plus en plus complexe et compétitif.

1

Prérequis théoriques et présentation de l'organisme d'accueil

Sommaire

Introduction 3
1.1 Présentation de Cevital 4
1.2 Situation géographique 4
1.3 Organigramme de Cevital 5
1.4 Missions du département cible : Reporting et analyse des ventes 6
1.5 Défis et contraintes 7
1.6 Problématique 7
1.7 Solution proposée 8
1.8 Principes des prévision des ventes 8
1.9 Fondements des séries chronologiques 10
1.10 Modèles d'analyse des séries chronologiques 13
1.11 Modèles de prévision et de modélisation 14
Conclusion 18

Introduction

Ce chapitre est consacré tout d'abord à la présentation de l'organisme d'accueil CEVITAL Agro-Alimentaire, nous explorerons l'histoire de l'entreprise, sa situation géographique stratégique, ainsi que son organigramme pour mieux comprendre son fonctionnement interne.

Ensuite, nous nous pencherons sur l'analyse prédictive et son importance dans le contexte commercial moderne. Nous discuterons des séries chronologiques, une méthode statistique clé

utilisée pour analyser les tendances temporelles dans les données. En particulier, nous évoquerons des méthodes de prévision spécifiques telles que ARIMA, SARIMA et Box-Jenkins. Ces méthodes sont couramment utilisées dans l'analyse de séries chronologiques pour comprendre les données et prédire les tendances futures.

L'objectif de ce chapitre est de fournir une vue d'ensemble de l'entreprise et d'établir les fondements pour une analyse détaillée de la pertinence et des techniques de prédiction dans les chapitres suivants.

1.1 Présentation de Cevital

Cevital Agro-Alimentaire : Bien plus qu'une entreprise, un moteur de développement.

Cevital Agro-Alimentaire, fondée en 1998 par M. Issaad Rebrab et ses enfants, est un leader du secteur privé de l'industrie agroalimentaire en Algérie. Cette entreprise familiale, connue pour son histoire et ses valeurs, joue un rôle crucial dans le développement économique national en répondant aux besoins locaux et en générant des excédents pour l'exportation.

Cevital propose une large gamme de produits alimentaires et de boissons de haute qualité, tels que des huiles, du sucre, de la margarine, des sauces, des jus de fruits, de l'eau minérale et de la chaux, à des prix compétitifs. La figure 1.1 représente le logo de Cevital et ses différents produits.

FIGURE 1.1 – Logo de l'entreprise Cevital et ses différentes marques de produits

Le groupe Cevital, comprenant 26 filiales dont Cevital Agro-Alimentaire, s'est bâti sur des valeurs solides, assurant sa réussite et sa renommée. Actif dans divers secteurs, il contribue significativement à la croissance économique et au développement social de l'Algérie.

1.2 Situation géographique

Cevital SPA : Un point d'ancrage stratégique au cœur de l'Algérie.

Située à Béjaïa, à l'extrémité du port et à proximité de la RN 26, Cevital SPA bénéficie d'une position géographique exceptionnelle.

D'une superficie de 45 000 m^2, le complexe est idéalement connectée aux réseaux de transport routier et maritime. Sa proximité du port, à seulement 3 km du centre-ville et à 200 mètres du quai, facilite l'acheminement des matières premières et des produits finis, conférant à l'entreprise un avantage économique considérable. La figure 1.2 représente la situation géographique du complexe Cevital.

FIGURE 1.2 – Situation géographique du complexe Cevital

1.3 Organigramme de Cevital

Afin de mieux cerner l'organisation interne du complexe Cevital, la figure 1.3 offre une vue d'ensemble des différents organes qui le constituent, illustrant ainsi les liens et la répartition des fonctions au sein de l'entité.

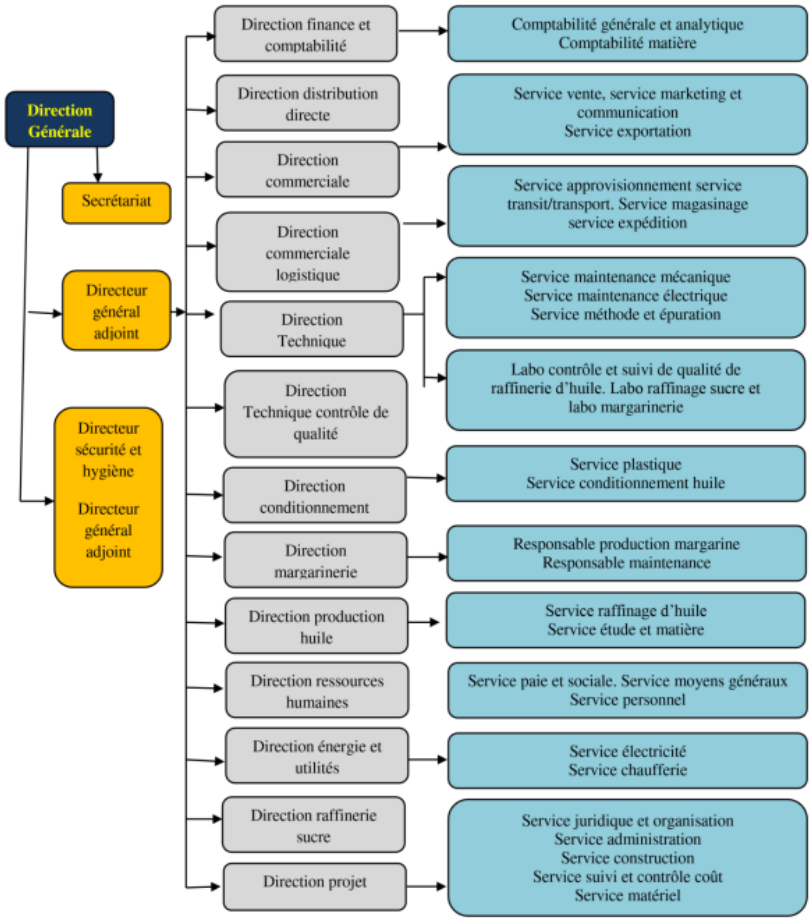

FIGURE 1.3 – Organigramme de Cevital

1.4 Missions du département cible : Reporting et analyse des ventes

Durant notre stage chez Cevital, nous avons eu l'opportunité d'intégrer le département Reporting et analyse des ventes. Cette immersion nous a permis d'observer de près les processus et les défis auxquels l'équipe est confrontée, et c'est de cette expérience que nous avons tiré la problématique de notre projet. Les missions centrales du département en question comprennent :

- Analyse approfondie des performances commerciales : Évaluation et analyse des ventes passées pour déceler les tendances, les opportunités et les défis.
- Prévision des ventes : Utilisation de méthodes statistiques et de modélisation, incluant l'apprentissage automatique, pour anticiper les ventes futures en se basant sur les données historiques et les variables pertinentes.
- Reporting et présentation des données : Élaboration de rapports réguliers et de tableaux de bord visuels afin de transmettre efficacement les performances commerciales aux parties prenantes internes et externes.
- Analyse de marché : Surveillance et analyse des tendances du marché, de la concurrence

et des comportements des consommateurs pour éclairer les prévisions et les stratégies commerciales.

- Optimisation des processus commerciaux : Identification et recommandation d'améliorations visant à optimiser les processus de vente, de distribution et de tarification pour accroître l'efficacité et la rentabilité.
- Collaboration interfonctionnelle : Travailler en étroite collaboration avec les équipes de vente, de marketing, de production et de finance pour aligner les prévisions de vente sur les objectifs commerciaux globaux.
- Formation et support : Offrir une formation et un soutien aux utilisateurs des données et des outils d'analyse pour garantir une utilisation efficace et une interprétation précise des informations commerciales.

1.5 Défis et contraintes

Cevital est confronté à plusieurs défis et contraintes majeurs qui compliquent l'analyse des ventes on peut citer :

- Diversité des produits et marchés : Cevital évolue dans une multitude de domaines, tels que l'agroalimentaire, les matériaux de construction et la distribution. Cette diversification des produits et des marchés complexifie les prévisions de ventes, en raison des facteurs spécifiques à chaque secteur et marché géographique.
- Facteurs externes : La volatilité des prix des matières premières, les fluctuations des taux de change, les évolutions réglementaires et d'autres facteurs externes tels que les aléas climatiques et les crises sanitaires peuvent influencer considérablement la demande pour les produits de Cevital. Ces éléments, souvent imprévisibles, ajoutent une incertitude significative aux prévisions de ventes.
- Systèmes et processus inefficaces : Le recours à des processus manuels ou à des systèmes de prévision des ventes obsolètes peut s'avérer inefficace et générer des erreurs. Il est crucial de mettre en place des systèmes et des processus automatisés capables de traiter de gros volumes de données et de produire des prévisions précises en temps réel.
- Concurrence et dynamique du marché : L'intensité de la concurrence, les changements des préférences des consommateurs et les évolutions du marché sont autant de facteurs susceptibles d'accroître l'incertitude des prévisions de ventes, particulièrement dans les secteurs hautement concurrentiels.

1.6 Problématique

Lors de notre stage chez Cevital, nous avons identifié plusieurs défis majeurs liés à la prévision des ventes. Actuellement, les prévisions sont principalement effectuées à l'aide d'outils traditionnels tels qu'Excel. Cette méthode, bien que familière, présente des limites en termes de précision et d'anticipation des fluctuations du marché.

Face à l'essor des technologies de l'information et de l'analyse de données, il est crucial d'explorer des méthodes avancées pour améliorer la précision des prévisions. L'automatisation

des prévisions est essentielle pour minimiser les erreurs et augmenter la fiabilité, particulièrement avec la croissance exponentielle des données (big data).

Cevital, avec sa large gamme de produits commercialisés à l'échelle nationale et internationale, nécessite des prévisions précises adaptées à chacun de ses produits. Nous avons pu recueillir des données sur différentes sous-familles d'un produit, désignées P1, P2 et P3, distribuées à travers divers canaux de ventes (C1, C2, C3) et dans différentes régions (A, B, C) sur une période de 5 ans.

Notre problématique se focalise donc sur la transition des méthodes classiques à des approches basées sur l'apprentissage automatique pour la prévision des ventes chez Cevital. Nous visons à démontrer que l'adoption de ces nouvelles technologies permettra non seulement d'améliorer la précision des prévisions mais aussi de fournir des insights plus approfondis sur les dynamiques du marché, facilitant ainsi une prise de décision plus éclairée et réactive.

1.7 Solution proposée

Ce projet s'inscrit dans une démarche d'innovation technologique au sein de Cevital, avec pour ambition de renforcer leur compétitivité en optimisant la gestion des ventes et en anticipant plus efficacement les demandes futures. Notre approche se basera exclusivement sur l'analyse des données historiques, cherchant à prouver que même sans données externes, les techniques avancées d'apprentissage automatique peuvent considérablement améliorer les prévisions de ventes.

Pour ce faire, nous allons explorer plusieurs modèles de prévision incluant SARIMA, forêts aléatoires, XGBoost, CatBoost, arbres de décision, SARIMA-CatBoost, etc. Les modèles seront évalués et comparés pour en choisir les plus performants pour les prévisions.

1.8 Principes des prévision des ventes

Dans cette section, nous explorerons les principes fondamentaux de la prévision des ventes, en soulignant l'importance cruciale de cette activité, ainsi que les différentes méthodes employées pour réaliser des prévisions précises et fiables.

1.8.1 Ventes

Les ventes constituent l'essence même de toute entreprise prospère. Elles représentent l'échange de biens ou de services contre une contrepartie financière, permettant ainsi de générer des revenus et de propulser la croissance de l'entreprise. Elles peuvent concerner une large gamme de produits ou de services, et elles impliquent généralement des activités telles que la prospection de clients, la promotion des produits, la négociation des conditions de vente, et la conclusion de transactions.

1.8.2 Prévision

La prévision est une démarche stratégique visant à anticiper les événements futurs en analysant les données passées. Utilisant diverses techniques, allant des méthodes statistiques clas-

siques aux technologies d'intelligence artificielle, elle s'adapte au phénomène étudié et aux préférences du chercheur. Son objectif est d'exploiter les informations disponibles pour éclairer les décisions futures et atteindre les objectifs spécifiques.

Bien que la prédiction parfaite soit souvent hors de portée, ces méthodes offrent des perspectives précieuses et des tendances générales pour guider les choix stratégiques. La prédiction se divise en trois horizons temporels : court terme, moyen terme et long terme, permettant d'adapter les approches en fonction de la durée visée [15].

1.8.3 Prévision des ventes

La prévision des ventes est un pilier essentiel pour les entreprises, leur fournissant une vision prospective de leurs activités en se basant sur des données historiques et actuelles ainsi que sur les conditions environnementales. Elle offre une représentation réaliste de ce que seront probablement les ventes à venir. Parmi les différentes définitions de la prévision des ventes, on trouve :

Définition 1.1. Yves Chirouze : « prévoir ses ventes consiste pour une entreprise à estimer par avance, pour un futur donné, le niveau de ses ventes compte tenu de ses actions commerciales, de son plan de marketing et des contraintes environnementales qu'elle pense subir. La prévision n'est ni une science exacte ni un art divinatoire. Prévoir nécessite une attitude scientifique qui suppose la collecte d'informations, leur analyse et pour certaines d'entre elles, un traitement à l'aide de méthodes spécialement mises au point » [9].

1.8.4 Importance et objectifs

Les prévisions de ventes jouent un rôle crucial dans la stratégie commerciale de toute entreprise, y compris Cevital. Elles permettent de :

- Anticipation de la demande : Ajuster la production en fonction de la demande, évitant ainsi les surplus de stocks ou les ruptures.
- Planification proactive : Allouer efficacement les ressources pour une meilleure efficacité opérationnelle.
- Vision stratégique : Anticiper les tendances du marché et identifier de nouvelles opportunités de croissance.
- Gestion financière : Établir des budgets précis et prévoir les besoins en capital.

1.8.5 Méthodes de la prévision des ventes

La prévision des ventes est essentielle pour anticiper les besoins du marché et optimiser les ressources. Elle s'appuie sur diverses méthodes qualitatives et quantitatives pour fournir des estimations précises et éclairer les décisions stratégiques [10].

Méthodes qualitatives

La méthode qualitative s'appuie sur le savoir et l'intuition d'experts internes à l'entreprise afin de formuler des prévisions sur un événement spécifique. Les experts sélectionnés sont re-

connus pour leur connaissance approfondie dans le domaine concerné et leur capacité à analyser des informations complexes. Elle se composent généralement de trois piliers [10].

- La méthode de sondage d'opinion : est largement employée pour recueillir des données auprès des vendeurs et des distributeurs de produits lors d'enquêtes.
- La méthode de comparaison : implique l'utilisation d'analogies historiques ou de comparaisons avec des produits similaires vendus dans le passé.
- La méthode de Delphi : implique la collecte des réponses de plusieurs experts via des questionnaires successifs. Son objectif est de mettre en lumière les convergences d'opinion et d'obtenir des consensus sur des sujets précis en interrogeant les experts de manière itérative.

Méthodes quantitatives

Celles-ci exploitent les données historiques et les techniques avancées d'analyse pour fournir des prévisions précises et des insights décisionnels. Voici quelques-unes des méthodes les plus couramment utilisées [10] :

- Analyse de séries chronologique : L'analyse de séries temporelles est une méthode puissante pour prévoir les ventes en capturant les tendances et les saisons dans les données historiques. Les techniques couramment utilisées incluent le lissage exponentiel, les modèles ARIMA et SARIMA (Seasonal ARIMA). Ces méthodes permettent de modéliser les variations temporelles et de générer des prévisions précises.
- Modèles de machine learning : Les techniques de machine learning telles que les forêts aléatoires et les réseaux de neurones peuvent également être utilisées pour prédire les ventes. Ces modèles sont capables de capturer des relations complexes entre les variables et de fournir des prévisions précises en utilisant des ensembles de données historiques.
- Méthodes de simulation : Les méthodes de simulation, comme celle de Monte Carlo, sont utilisées pour générer des scénarios futurs possibles en prenant en compte l'incertitude et la variabilité des données. Cela permet d'explorer différentes hypothèses et de prendre des décisions éclairées en matière de gestion des ventes.

Parmi les méthodes de prévision des ventes évoquées précédemment, notre étude se focalisera principalement sur l'application des techniques de Machine Learning.

Avant d'approfondir cette approche, nous commencerons par explorer les fondements des séries chronologiques, essentiels pour comprendre le contexte et les données sur lesquels reposent nos modèles prédictifs.

1.9 Fondements des séries chronologiques

Les séries temporelles sont une extension naturelle des méthodes de prévision, permettant l'étude de l'évolution des valeurs numériques collectées à intervalles réguliers. Elles aident à détecter les tendances, saisonnalités et cycles dans les données. Intégrer les séries temporelles dans notre boîte à outils d'analyse prédictive améliore notre compréhension des modèles temporels et augmente la précision de nos prévisions [15].

Définition 1.2. Un processus stochastique est une famille $X = \{X_t : t \in \mathbb{T}\}$ de variables aléatoires indexées par $\mathbb{T}$ et définies sur un même espace probabilisé $(\Omega, \mathcal{F}, \mathbb{P})$ et à valeurs $(\mathbb{R}, \mathcal{B}_{\mathbb{R}})$.

Définition 1.3. Une série chronologique ou série temporelle est un processus stochastique $\{X_t : t \in \mathbb{Z}\}$ tel que $X_t \in L^2(\Omega, \mathcal{F}, \mathbb{P})$. Elle se compose d'une séquence de variables aléatoires. Le paramètre t représentant le temps. La taille de cette série est désignée par le nombre n, dénommé la longueur de la série [13].

Séries chronologiques stationnaires

Une série temporelle est dite stationnaire lorsqu'elle ne présente ni tendance, ni saisonnalité, ni aucune évolution temporelle de ses propriétés statistiques.

Un processus stochastique $X = \{X_t : t \in \mathbb{T}\}$ est considéré comme stationnaire lorsque ses moments du premier et du second ordre restent invariants dans le temps, quelle que soit la période considérée, ce qui se traduit par les conditions suivantes [5] :

1. **Espérance** : L'espérance de X_t demeure constante et finie pour tout $t \in \mathbb{T}$, exprimée par :

$$\mathbb{E}(X_t) = \mu < \infty. \tag{1.1}$$

2. **Variance** : La variance de X_t est constante et finie pour tout $t \in \mathbb{T}$, ce qui est notée par :

$$\mathrm{Var}(X_t) = \sigma^2 < \infty. \tag{1.2}$$

3. **Covariance :** La covariance entre X_t et X_{t-k}, où k représente un décalage temporel, demeure constante et dépend uniquement de la différence temporelle k, notée par :

$$\mathrm{Cov}(X_t, X_{t-k}) = E[X_t - \mu)(X_{t-k} - \mu)] = \gamma_k \tag{1.3}$$

pour tous $t, k \in \mathbb{T}$.

Test de stationnarité

Pour tester la stationnarité d'une série chronologique, il existe plusieurs, on peut citer le test de Dickey-Fuller Augmenté (ADF) :

Test de Dickey-Fuller augmenté (ADF) : C'est un test statistique populaire qui détecte la présence d'une racine unitaire, indiquant si une série temporelle est stationnaire ou non, en vérifiant la significativité de sa composante autoregressive [5]. L'équation du test de Dickey-Fuller augmenté est donnée par :

$$\Delta X_t = \alpha + \beta t + \gamma X_{t-1} + \sum_{i=1}^{p} \delta_i \Delta X_{t-i} + \varepsilon_t \tag{1.4}$$

Dans cette formule :

- $\Delta X_t := X_t - X_{t-1}$, elle représente la différence de la série temporelle à l'instant t.

- α est une constante.
- βt est la tendance temporelle.
- $\gamma\, X_{t-1}$ est le terme de retard de la série temporelle.
- $\delta_i\, \Delta\, X_{t-i}$ sont les termes de retard des différences de la série temporelle.
- (ε_t) est un bruit blanc.

Définition 1.4. Un processus $\{\epsilon_t : t \in \mathbb{Z}\}$ est un bruit blanc (faible) si $\mathbb{E}\epsilon_t = 0$, $\mathbb{E}\epsilon_t^2 = \sigma^2$, $\mathrm{Cov}(\epsilon_s, \epsilon_t) = 0$ pour $s \neq t$.

Décomposition saisonnière

Pour analyser efficacement une série chronologique, il est crucial de décomposer sa structure afin de mieux cerner ses tendances et variations. En identifiant et en modélisant séparément ses différentes composantes, on obtient une meilleure compréhension des données, des prévisions plus précises et des outils précieux pour la prise de décision [5].

- **La tendance (trend)** (Z_t) : La tendance représente l'évolution à long terme de la série chronologique. Elle capture le mouvement général des données, en filtrant les variations à court terme. La tendance peut être linéaire, exponentielle, polynomiale ou encore suivre un modèle plus complexe.
- **La saisonnalité** (S_t) : La saisonnalité représente les variations cycliques qui se répètent à intervalles réguliers, souvent liés à des facteurs saisonniers comme les mois de l'année ou les jours de la semaine. Elle est particulièrement importante dans des domaines comme la vente au détail ou le tourisme.
- **Les résidus** (ϵ_t) : Les résidus, aussi appelés bruit ou aléas, représentent les fluctuations irrégulières et imprévisibles de la série chronologique. Ils regroupent les événements ponctuels et les effets aléatoires qui ne peuvent être expliqués par les composantes tendance et saisonnalité.
- **Les événements accidentels** (R_t) : Des événements exceptionnels, tels que des grèves, des catastrophes naturelles ou des crises financières, peuvent également influencer l'évolution d'une série chronologique. Ces événements sont généralement intégrés dans la composante résiduelle, mais peuvent parfois nécessiter un traitement distinct si leur impact est significatif.
- **Le cycle** (C_t) : Représente la composante cyclique à un moment donné t. Celle-ci capture les fluctuations périodiques qui se répètent à intervalles réguliers dans la série temporelle, mais qui ne sont pas liées à des variations saisonnières.

La figure 1.4 correspond aux composantes d'une série chronologique.

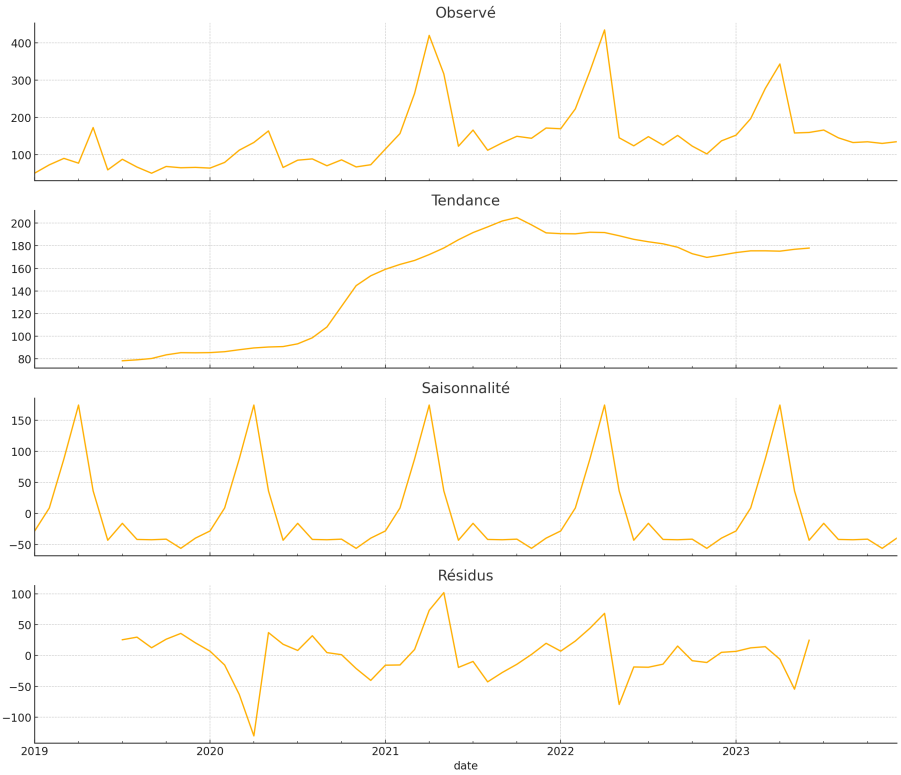

FIGURE 1.4 – Composantes d'une série chronologique

Au-delà de la décomposition classique des séries temporelles, d'autres méthodes d'analyse enrichissent leur compréhension. L'analyse des ruptures identifie les points de changement brutaux. L'analyse de causalité explore les relations de cause à effet entre variables. L'analyse de survie évalue le temps avant la survenue d'événements spécifiques. Ces approches complémentaires offrent une diversité de perspectives et d'outils pour mieux interpréter les données temporelles, aidant ainsi à la prise de décision et la planification stratégique.

1.10 Modèles d'analyse des séries chronologiques

Dans l'univers complexe de l'analyse des séries chronologiques, les modèles additifs, multiplicatifs et mixtes se distinguent comme des outils puissants, offrant des perspectives uniques pour déchiffrer les mystères des tendances, des cycles et des variations saisonnières qui se cachent dans les données temporelles [5].

- Modèle additif : Représente une façon de décomposer une série temporelle où la tendance, la saisonnalité et les résidus sont simplement additionnés pour former la série observée. Ce schéma convient lorsque les différentes composantes sont considérées comme indépendantes et que leurs effets s'ajoutent. Sa formule est donnée comme suit :

$$X_t = Z_t + S_t + C_t + \varepsilon_t, \quad t \in \mathbb{Z}. \tag{1.5}$$

- Modèle multiplicatif : Ce modèle est préférable lorsque les effets des différentes composantes sont considérés comme interdépendants et sont mieux représentés par une combinaison multiplicative.

Dans ce modèle, une série chronologique est exprimée de la manière suivante :

$$X_t = Z_t \cdot S_t \cdot C_t \cdot \varepsilon_t, \quad t \in \mathbb{Z}. \tag{1.6}$$

- Modèle mixte : Dans ce schéma , les composantes de la série chronologique sont combinées à la fois par des opérations d'addition et de multiplication. Ce dernier est préféré lorsqu'il est nécessaire de représenter de manière optimale les interactions entre les différentes composantes, tant par des ajouts que par des multiplications. Sa formule est donnée comme suit :

$$X_t = (Z_t + C_t) \cdot S_t + \varepsilon_t, \quad t \in \mathbb{Z}. \tag{1.7}$$

Pour déterminer si une série chronologique suit un modèle additif, multiplicatif ou mixte, plusieurs approches peuvent être employées :

- Méthode graphique : Pour modéliser la saisonnalité d'une série temporelle, deux approches existent : additive si les variations saisonnières ont une amplitude constante, ou multiplicative si elles évoluent proportionnellement à la tendance. Dans certains cas, une approche mixte combinant les deux modèles peut aussi être envisagée [5]. Les modèles graphiques sont représentés dans la figure 1.5.

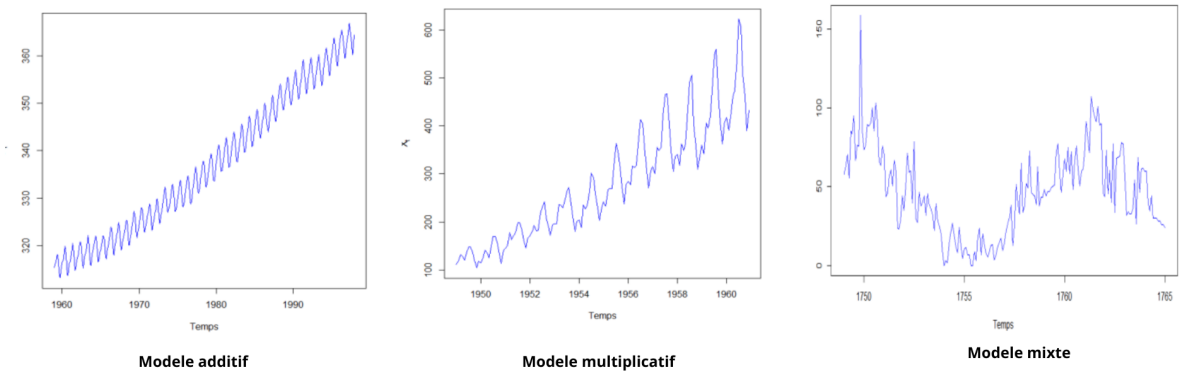

FIGURE 1.5 – Représentation des modèles graphiques

- Méthode analytique : Des tests statistiques, comme le test de Buys-Ballot ou le test de Kuiper, peuvent être appliqués pour déterminer formellement le type de modèle. Ces tests s'appuient sur l'analyse de la variance des résidus du modèle [5].

1.11 Modèles de prévision et de modélisation

Les analystes utilisent diverses méthodes pour anticiper de manière précise les tendances futures. Parmi les approches couramment utilisées, on compte le lissage exponentiel, les moyennes mobiles MA, les modèles autorégressifs AR, les modèles ARIMA, les modèles SARIMA ainsi que la méthodologie de Box-Jenkins. Ces méthodes sont utilisées dans la planification,l'élaboration des politiques, la prise de décision et la prédiction [15].
Nous mentionnons quelques-unes dans ce qui suit :

1.11.1 Modèle auto-régressif AR(p)

Le modèle auto-régressif de degré p, souvent noté AR(p) ou ARIMA(p, 0, 0), est largement utilisé dans la prédiction des variables macroéconomiques. Il est basé sur l'idée que les valeurs passées de la série chronologique peuvent être utilisées pour prédire les valeurs futures, avec des poids ϕ_i attribués à chaque valeur passée. On dit qu'une série chronologique $X = \{X_t : t \in \mathbb{Z}\}$ suit le modele AR(p) si :

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \cdots + \phi_p X_{t-p} + \epsilon_t, \tag{1.8}$$

avec :

- $\phi_1, \phi_2, ..., \phi_p \in \mathbb{R}$, $\phi_p \neq 0$, sont les coefficients d'auto-régression qui mesurent l'impact des valeurs passées de la série sur la valeur actuelle X_t. Chaque coefficient ϕ_i indique l'importance de la valeur de la série à l'instant $t - i$ sur X_t.
- (ϵ_t) est un bruit blanc qui capture les variations non expliquées par le modèle.

1.11.2 Moyennes mobiles MA(q)

Les moyennes mobiles sont utilisées pour modéliser des données saisonnières ou périodiques avec un nombre limité d'observations. Ce modèle accorde plus d'importance aux observations les plus récentes dans la prédiction en calculant la moyenne des n dernières valeurs observées pour estimer la prochaine valeur. On dit qu'une série chronologique $X = \{X_t : t \in \mathbb{Z}\}$ suit le modèle MA(q) si :

$$X_t = \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \cdots + \theta_q \epsilon_{t-q}, \tag{1.9}$$

où :

- $\theta_1, \theta_2, \ldots, \theta_q, \in \mathbb{R}$, $\theta_p \neq 0$ sont les coefficients des termes de moyenne mobile.
- (ϵ_t) est un bruit blanc.

1.11.3 Modèles ARMA(p,q) et ARIMA(p,d,q)

ARMA (AutoRegressive Moving Average)

ARMA est un modèle qui combine des composantes auto-régressives (AR) et de moyennes mobiles (MA). Les termes AR capturent la dépendance des observations actuelles par rapport aux observations passées, tandis que les termes MA capturent la dépendance des erreurs de prédiction passées. il est utilisé pour les séries stationnaires.

On dit qu'une série chronologique $X = \{X_t : t \in \mathbb{Z}\}$ suit le modèle ARMA(p, d) si :

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \ldots + \phi_p X_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + \theta_2 \epsilon_{t-2} + \ldots + \theta_q \epsilon_{t-q}, \tag{1.10}$$

- $\phi_1, \phi_2, ..., \phi_p \in \mathbb{R}$, $\phi_p \neq 0$, sont les coefficients d'auto-régression qui mesurent l'impact des valeurs passées de la série sur la valeur actuelle X_t. Chaque coefficient ϕ_i indique l'importance de la valeur de la série à l'instant $t - i$ sur X_t.
- $\theta_1, \theta_2, \ldots, \theta_q, \in \mathbb{R}$, $\theta_p \neq 0$ sont les coefficients des termes de moyenne mobile.

- (ϵ_t) est un bruit blanc.

Les paramètres (p,q) correspondent aux :
p : ordre autoregressif (AR).
q : ordre de la moyenne mobile (MA).

ARIMA (Autoregressive integrated moving average)

ARIMA est une extension du modèle ARMA qui incorpore également une composante d'intégration I. Avant d'appliquer le modèle, la série chronologique est différenciée d fois pour la rendre stationnaire, éliminant ainsi les tendances et les structures saisonnières. Le paramètre d supplémentaire correspond à l'ordre de différenciation.

1.11.4 Modèle SARIMA $((p, d, q) \times (P, D, Q))_s$

Le modèle SARIMA (seasonal ARIMA) est une extension du modèle ARIMA qui prend en compte la saisonnalité. Il permet de modéliser les variations cycliques qui se répètent à intervalles réguliers, comme les variations mensuelles, trimestrielles ou annuelles. On dit qu'une série chronologique $X = \{X_t : t \in \mathbb{Z}\}$ suit le modèle SARIMA $((p, d, q) \times (P, D, Q))_s$ si :

$$X_t = \sum_{i=1}^{p} \phi_i X_{t-i} + \sum_{j=1}^{q} \Theta_j \epsilon_{t-j} + \sum_{k=1}^{P} \Phi_k \epsilon_{t-ks} + \sum_{l=1}^{Q} \Theta_l^* \epsilon_{t-ls} + \delta - (1-B)^d X_t - (1-B^s)^D X_t, \quad (1.11)$$

Les différents termes et coefficients de cette équation sont expliqués ci-dessous :

- B : L'opérateur de retard (ou décalage), qui déplace les valeurs de la série dans le temps (par exemple, $BX_t = X_{t-1}$).
- ϕ_i : Coefficients autorégressifs non-saisonniers.
- Θ_j : Coefficients de moyennes mobiles non-saisonniers.
- Φ_k : Coefficients autorégressifs saisonniers.
- Θ_l^* : Coefficients de moyennes mobiles saisonniers.
- δ : Une constante qui peut être ajoutée au modèle.
- $(1 - B)^d X_t$: La différenciation non-saisonnière d'ordre d, utilisée pour rendre la série stationnaire en éliminant les tendances.
- $(1 - B^s)^D X_t$: La différenciation saisonnière d'ordre D, utilisée pour rendre la série stationnaire en éliminant les tendances saisonnières.
- (ϵ_t) est un bruit blanc.
- p, d, q sont les ordres de la composante non saisonnière de l'ARIMA.
- P, D, Q sont les ordres de la composante saisonnière de l'ARIMA.
- s est la composante de saisonnalité.

1.11.5 Box-Jenkins

Nommé en l'honneur des deux statisticiens qui l'ont développé et initiée en 1976, Box-Jenkins est une méthode de régression automatique et de moyenne mobile largement utilisée pour la prévision des séries chronologiques. Elle est structurée autour de plusieurs étapes clés qui guident le processus d'analyse et de prévision. Dans cette méthodologie, nous pouvons distinguer les trois principales, suivies par un schéma détaillant les six étapes complètes.

1. Identification du modèle :
   - Analyse exploratoire des données.
   - Sélection du modèle ARIMA initial en utilisant les graphiques de la série temporelle, les ACF et les PACF.
2. Estimation des paramètres du modèle :
   - Utilisation de techniques statistiques pour estimer les paramètres du modèle ARIMA, tels que la méthode des moindres carrés ou la maximisation de la vraisemblance.
3. Validation du modèle :
   - Évaluation de la qualité du modèle en examinant les résidus pour la stationnarité, l'autocorrélation et la normalité [13].

La figure 1.6 représente les étapes de la méthodologie de Box-Jenkins.

FIGURE 1.6 – Étapes de la méthodologie Box-Jenkins

Conclusion

Pour conclure, comprendre et analyser les séries chronologiques est essentiel pour prendre des décisions éclairées. Ce chapitre a exploré en détail les aspects théoriques de la prévision des ventes, les fondements des séries chronologiques, en mettant l'accent sur les concepts clés tels que ses différentes composantes, nous avons également examiné différentes méthodes de modélisation, notamment les approches classiques telles que le modèle SARIMA.

Comprendre ces bases et maîtriser ces concepts est essentiel pour interpréter et anticiper les tendances dans divers domaines. Dans le prochain chapitre, nous explorerons une approche complémentaire et innovante pour modéliser les séries chronologiques, en nous tournant vers l'apprentissage automatique. Cette méthode élargit nos horizons en permettant aux algorithmes d'extraire des modèles complexes directement à partir des données, ouvrant ainsi de nouvelles perspectives pour une analyse plus approfondie et des prévisions plus précises.

En conclusion, ce chapitre a mis en lumière les aspects théoriques et pratiques de la prévision des ventes, en se concentrant sur l'exemple de Cevital. Nous avons abordé les principales missions et défis de l'entreprise, et proposé des solutions basées sur des méthodes de prévision avancées. Les fondements des séries chronologiques et les différents modèles de prévision, tels que les modèles ARIMA, ont été discutés en détail pour montrer leur pertinence et leur application dans le cadre de la gestion des ventes. L'objectif était de démontrer comment une approche méthodique et scientifique peut améliorer la précision des prévisions et, par conséquent, l'efficacité opérationnelle de l'entreprise. Cette analyse pose les bases pour les chapitres suivants, où nous explorerons l'application de l'apprentissage automatique et d'autres techniques avancées pour affiner encore davantage les prévisions des ventes.

2
Apprentissage automatique

Sommaire

Introduction **19**
2.1 Intelligence artificielle **20**
2.2 Apprentissage automatique **20**
2.3 Types d'apprentissage automatique **24**
2.4 Algorithmes de l'apprentissage automatique **27**
2.5 Apprentissage profond **40**
2.6 Optimisation et validation des modèles **40**
2.7 Métriques d'évaluation des modèles **41**
2.8 Travaux connexes **42**
Conclusion **48**

Introduction

L'avènement de l'intelligence artificielle et de l'apprentissage automatique a marqué une transformation radicale dans le traitement et l'analyse des données. Ce chapitre explore l'univers de l'apprentissage automatique, une sous-discipline de l'IA qui permet aux systèmes d'apprendre et de s'améliorer automatiquement à partir de l'expérience sans être explicitement programmés.

Nous commencerons par définir l'intelligence artificielle et l'apprentissage automatique avant d'explorer ses divers types, ses principaux algorithmes et leurs applications dans la prédiction des données. Ensuite, une attention particulière sera accordée à l'optimisation et à la validation des modèles pour garantir l'exactitude et la rigueur des prévisions. Nous discuterons des métriques d'évaluation couramment utilisées afin de garantir des prévisions précises et fiables. Nous finirons par passer en revue les travaux connexes dans le domaine.

2.1 Intelligence artificielle

L'intelligence artificielle (IA), vise à doter les machines d'une aptitude naturelle pour penser, raisonner et trouver des solutions semblables aux défis que rencontrent les individus. La résolution des problèmes est fréquemment considérée comme un indicateur important de l'intelligence. Malgré cela, la définition formelle de l'intelligence continue d'être un défi complexe.

Le concept fondamental repose sur la création d'entités ayant une intelligence très développée afin qu'elles puissent effectuer diverses tâches complexes telles que la reconnaissance ou le processus décisionnel autonome. Dans cette quête, les chercheurs explorent maintes différentes approches pour résoudre le problème. À la fois les règles explicites, les heuristiques et les techniques de l'apprentissage automatique sont comprises dans cela [19]. La figure 2.1 illustre l'écosystème de l'intelligence artificielle.

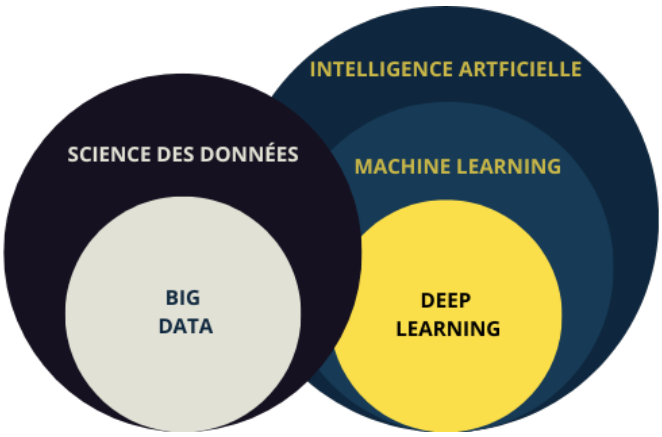

FIGURE 2.1 – Panorama de l'intelligence artificielle

2.2 Apprentissage automatique

L'apprentissage automatique (Machine Learning), une branche de l'intelligence artificielle, utilise des algorithmes pour apprendre et s'améliorer à partir de données. Ces algorithmes permettent aux ordinateurs d'accomplir des tâches complexes pour aider à la prise de décisions. L'apprentissage automatique vise à développer des solutions innovantes dans divers secteurs [3].

L'apprentissage automatique, défini par Arthur Samuel un informaticien d'IBM en 1959 [3], est l'acquisition de connaissances sans programmation explicite. Utilisant des outils statistiques et des algorithmes, il crée automatiquement des modèles à partir de données d'apprentissage. Ces modèles peuvent ensuite prédire de nouvelles données et effectuer des analyses avec une intervention humaine minimale. Ils sont cruciaux pour des applications réelles comme la vision par ordinateur, l'aérospatiale, l'économie, la santé, les médias et la gestion de la chaîne d'approvisionnement [6].

Modèle d'apprentissage

Considérons un ensemble de données $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, où x_i représente les caractéristiques (entrées) et y_i représente les étiquettes (sorties) correspondantes. Le but de l'apprentissage automatique est de trouver une fonction f telle que :

$$f(x_i) \approx y_i \quad \text{pour tous les } i. \tag{2.1}$$

La fonction f doit donc minimiser l'écart entre ses prédictions et les valeurs réelles y_i [19].

Fonction de perte

Pour mesurer l'efficacité de la fonction f trouvée par l'algorithme d'apprentissage, nous utilisons une fonction de perte. Cette dernière, notée $L(f(x), y)$, quantifie l'écart entre la prédiction $f(x)$ et la valeur réelle y. Une fonction de perte permet de transformer le problème d'apprentissage en un problème d'optimisation : en minimisant cette fonction, nous améliorons la précision de nos prédictions [19].

Pour la classification, où l'objectif est de prédire des étiquettes de classes discrètes, une fonction de perte courante est la log-loss, aussi connue sous le nom de perte logarithmique. La log-loss mesure la performance d'un modèle de classification dont la sortie est une probabilité comprise entre 0 et 1. Elle est définie comme suit :

$$L(f(x), y) = -[y \log(f(x)) + (1 - y) \log(1 - f(x))] \tag{2.2}$$

Dans cette formule, y est la véritable étiquette de classe (0 ou 1), et $f(x)$ est la probabilité prédite que x appartienne à la classe 1. La log-loss pénalise les prédictions incorrectes en attribuant une perte élevée lorsque la probabilité prédite est éloignée de la véritable étiquette. Cela encourage le modèle à attribuer des probabilités plus élevées aux classes correctes, améliorant ainsi la précision des prédictions de classification.

Objectif d'optimisation

Le but de l'apprentissage automatique est de minimiser la perte totale sur l'ensemble de données, ce qui se traduit par un problème d'optimisation. Concrètement, nous cherchons à trouver une fonction f qui minimise la somme des pertes pour toutes les paires de données (x_i, y_i) :

$$\min_f \sum_{i=1}^{n} L(f(x_i), y_i) \tag{2.3}$$

Afin de résoudre ce problème, plusieurs techniques peuvent être utilisées, comme la descente de gradient, où les paramètres du modèle sont ajustés iterativement pour minimiser la fonction de perte [19].

Descente de gradient

Actuellement,la descente de gradient se positionne comme la stratégie d'optimisation la plus couramment utilisée en ML et en DL. Elle est employée durant la phase de création de modèles de données, s'adapte à tous les algorithmes, et se révèle simple à comprendre et à implémenter.

La descente de gradient est un algorithme itératif d'optimisation visant à minimiser une fonction coût/perte en ajustant les paramètres d'un modèle. Bien d'efficace pour les fonctions convexes avec convergence vers le minimum global, elle est également appliquée à des fonctions non convexes où elle peut converger vers des minimums locaux. Son objectif est d'optimiser les performances du modèle [14].

Fonctionnement de la descente de gradient

Pour mieux comprendre le fonctionnement de la descente de gradient, considérons sa formule générale [14] :

$$x_{t+1} = x_t - \eta \nabla x_t \tag{2.4}$$

- x_{t+1} représente la valeur de x à l'itération suivante,
- x_t est la valeur actuelle de x,
- η (eta) est le taux d'apprentissage, qui détermine la taille du pas de mise à jour,
- ∇x_t est la direction de descente.

L'objectif de la descente de gradient est de minimiser la fonction objectif $f(x)$ en garantissant qu'à chaque itération, la valeur de f diminue, c'est-à-dire que $f(x_{t+1}) \leq f(x_t)$.

Description de l'algorithme

Algorithme 1 : Algorithme de descente de gradient

Données : Une fonction f et son gradient ∇f
Résultat : Un point x qui minimise f

1 **Initialisation** : Définir un point de départ x_0 dans le domaine de la fonction f;
2 **Itérations : Tant que** *le critère d'arrêt n'est pas satisfait* **faire** *Calculer la valeur de la fonction f au point x_t;*
3 *Mettre à jour les coordonnées de x en suivant la direction de descente $-\eta \nabla f(x_t)$;*
4 $x_{t+1} \leftarrow x_t - \eta \nabla f(x_t)$;
5 **Critère d'arrêt :** L'algorithme s'arrête lorsque l'un des critères suivants est satisfait :
- Différence de valeurs : $|f(x_{t+1}) - f(x_t)| \leq \varepsilon$, où ε est un petit nombre indiquant une faible variation de la fonction objectif.
- Nombre d'itérations : Limiter le nombre d'optimisations à un nombre prédéfini.

Explication détaillée

- **Gradient** $\nabla f(x_t)$: Le gradient de la fonction f en x_t est un vecteur indiquant la direction de la plus forte augmentation de f. En suivant la direction opposée au gradient, $-\nabla f(x_t)$, on se déplace vers la diminution la plus rapide de f.
- **Taux d'apprentissage** η : Le taux d'apprentissage détermine la taille du pas de mise à jour. Un taux trop élevé peut rendre la convergence instable, tandis qu'un taux trop faible peut ralentir la convergence.

- **Objectif** : L'objectif de chaque itération est de trouver une nouvelle valeur de x telle que la valeur de la fonction f diminue. Cela permet de s'approcher progressivement du minimum global de la fonction.

Le taux d'apprentissage est un élément essentiel pour accélérer la recherche du minimum global permet d'éviter les pièges des minima locaux. La figure 2.2 illustre un exemple d'un minimum, maximum global et local.

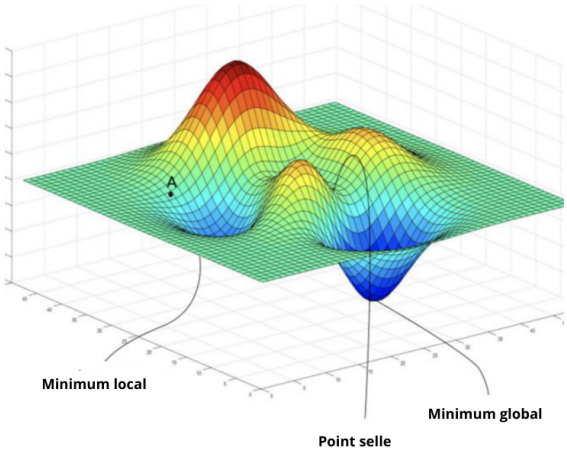

FIGURE 2.2 – Descente de gradient

2.2.1 Phases de l'apprentissage automatique

On distingue deux types de phases de l'apprentissage automatique [6].

- **La phase d'apprentissage :** Aussi appelée entraînement, une procédure cruciale. Elle expose le modèle à divers exemples significatifs pour qu'il apprenne les règles sous-jacentes en se basant sur les données d'entraînement. Cette étape est essentielle pour qu'il acquière les connaissances nécessaires.
- **La phase de test :** Également connue telle que phase de déploiement, le modèle utilise ces connaissances pour effectuer des prédictions ou prendre des décisions sur de nouvelles données dans des situations réelles même si elles n'ont pas été vues lors de l'apprentissage, démontrant ainsi sa capacité de généralisation.

La figure 2.3 décrit les différentes phases du ML.

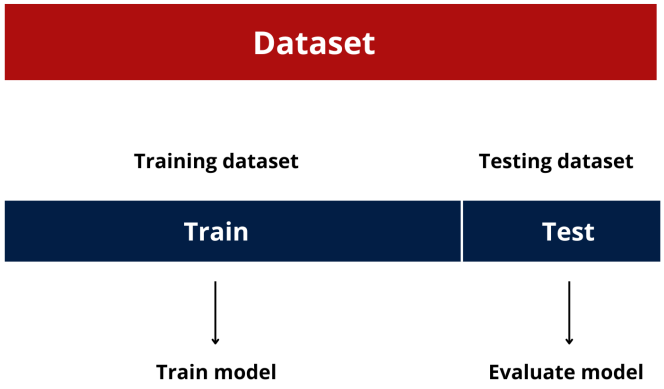

FIGURE 2.3 – Phases de l'apprentissage automatique

2.2.2 Limites de l'apprentissage automatique

L'apprentissage automatique offre des applications diverses et étendues, mais il présente également plusieurs limites à prendre en compte [6] :

- La précision des prédictions n'est jamais absolue, ce qui peut entraîner des erreurs dans certaines situations critiques.
- La complexité des algorithmes peut rendre leur fonctionnement opaque, compliquant ainsi la compréhension du processus décisionnel.
- La correction des erreurs et la détection d'anomalies peuvent être difficiles, impactant le développement et l'entraînement des modèles.
- La complexité exponentielle des algorithmes peut nécessiter des compromis entre efficacité et précision.
- Les performances des modèles dépendent fortement des données d'entraînement, nécessitant des ensembles de données représentatifs.
- Le fléau de la dimension peut rendre la gestion de données de grande dimension problématique, nécessitant l'utilisation de méthodes de réduction de dimensionnalité.

2.3 Types d'apprentissage automatique

Il existe plusieurs types d'apprentissage automatique, parmi lesquels se distinguent les quatre types les plus courants, comme illustré dans la figure 2.4.

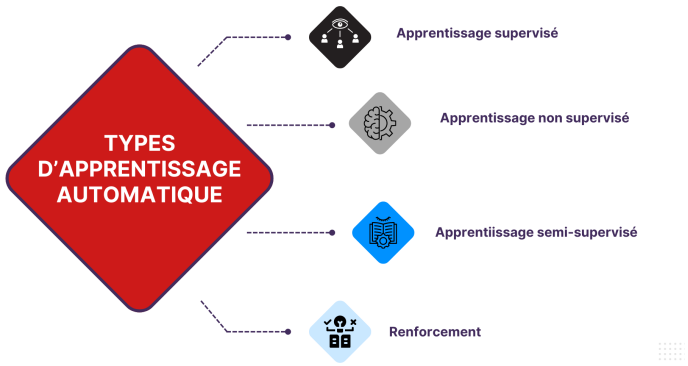

FIGURE 2.4 – Types d'apprentissage automatique

2.3.1 Apprentissage supervisé

L'apprentissage supervisé est une méthode où un algorithme est formé à partir de données d'entraînement étiquetées, composées d'exemples authentiques, préalablement traités et validés. L'objectif est d'identifier des liens significatifs entre les variables explicatives (données d'entrée) et les variables à prédire (données de sortie). Chaque instance est formalisé sous la forme d'un couple d'entrée-sortie (x_n, y_n) où x_n appartient à l'ensemble X des attributs (discrets ou continus), et y_n appartient à l'ensemble Y des valeurs de sortie pouvant être discrètes ou continues, également connu sous le nom de variable cible ou dépendante. Ce dernier forme les fondations de ce processus. L'apprentissage supervisé cherche ainsi à élaborer une fonction $f : X \rightarrow Y$ entre les variables prédictives en entrée x et la variable à prédire y, connue sous le nom de modèle de prédiction [3]. Les algorithmes d'apprentissage supervisé conviennent aux tâches suivantes :

- **Classification** : Lorsque la valeur à prédire est discrète. On distingue deux types de classification.
  1. Classification binaire : pour dissocier les données en deux catégories.
  2. Classification multi-classes : afin de classer les données en plusieurs catégories différentes
- **Régression** : Lorsque la valeur cible est continue. Il existe plusieurs modèles pour la régression.
  1. Régression linéaire : Modèle pour établir une relation linéaire entre les variables.
  2. Régression polynomiale : Modèle étendant la régression linéaire pour capturer des relations non linéaires.
  3. Régression logistique : Modèle spécifique pour la classification basée sur une fonction logistique.
- **Assemblage** : Est une agrégation des prédictions de multiples modèles d'apprentissage automatique en vue de générer une prédiction plus précise.

La figure 2.5 représente un exemple des tâches de l'apprentissage supervisé

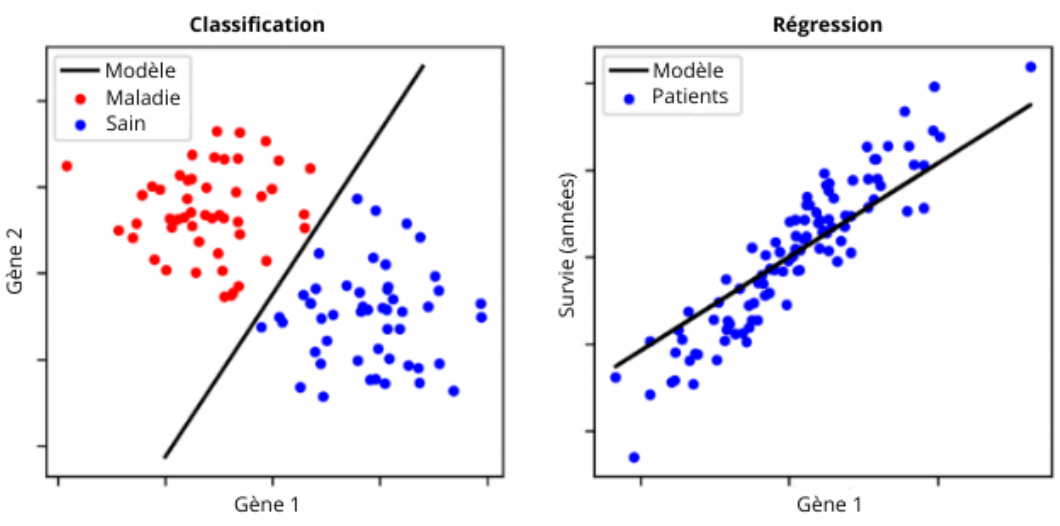

FIGURE 2.5 – Exemple des tâches de l'apprentissage supervisé

2.3.2 Apprentissage non supervisé

Pour cet apprentissage, les données ne sont pas étiquetées. L'algorithme analyse ces données pour découvrir automatiquement des modèles, des structures et des patterns cachés.

Cela permet à la machine de trouver des similarités et des relations entre les données par lui-même, ce qui peut être utile pour regrouper des éléments hétérogènes sous forme de sous groupes homogènes ou identifier des tendances dans de grands ensembles de données [3]. La figure 2.6 représente l'apprentissage non supervisé.

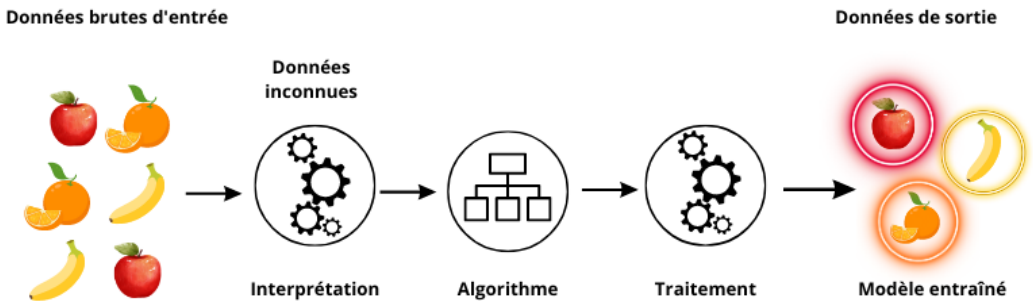

FIGURE 2.6 – Exemple d'apprentissage automatique non supervisé

Les algorithmes de cet apprentissage sont adaptés aux :

- **Clustering** : Division des données en clusters (groupes) selon leur similitude.
- **Détection d'anomalies** : Identification de points de données inhabituels dans un dataset.
- **Réduction de la dimensionnalité** : Diminution du nombre de variables dans un dataset.

- **Exploration d'associations** : Recherche de groupes d'éléments dans un ensemble de données fréquemment liés.

2.3.3 Apprentissage semi-supervisé

Ce dernier combine des données étiquetées et non étiquetées pour former des modèles robustes et plus précis tout en minimisant les coûts d'étiquetage et en économisant du temps. L'apprentissage semi supervisé exploite principalement des données non étiquetées, ce qui réduit les biais humains. Il est utilisé dans plusieurs domaines tels que la reconnaissance faciale, la détection des fraudes [3].

2.3.4 Apprentissage par renforcement

L'apprentissage par renforcement, quant a lui se fait par interaction de manière itérative entre un agent et son environnement, cela dit sans supervision. À travers l'expérience acquise par essai et erreur, en explorant différentes actions et en observant les récompenses ou pénalités associées à ces actions, l'agent ajuste sa stratégie pour maximiser les récompenses cumulées au fil du temps. L'apprentissage revient à établir une stratégie performante pour atteindre des objectifs spécifiques dans des environnements dynamiques et incertains. Cette approche est appliquée dans le domaine de la robotique, les jeux vidéos et les systèmes de recommandation et autre [3]. La figure 2.7 fait référence à l'apprentissage par renforcement.

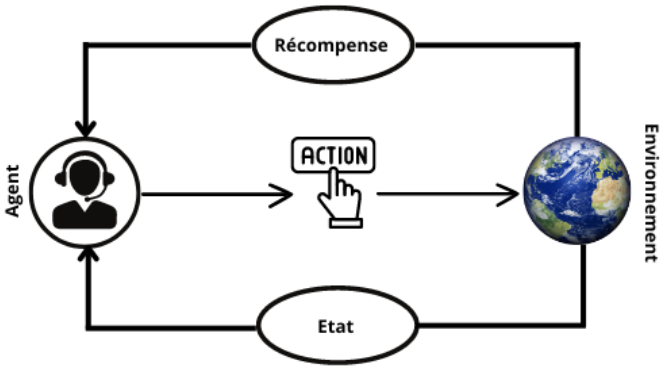

FIGURE 2.7 – Apprentissage par renforcement

2.4 Algorithmes de l'apprentissage automatique

Pour comprendre l'éventail des techniques d'apprentissage automatique, il est essentiel d'explorer ses différents algorithmes que nous récapitulons comme suit dans le tableau 2.1 [7].

Type d'apprentissage	Tâches	Algorithmes
Supervisé	Classification	Régression logistique, arbres de décision, SVM, KNN, Réseaux de neurones et la classification naïve bayésienne.
	Régression	Régression linéaire, régression ridge, régression lasso, réseaux de neurones.
	Assemblage	Bagging, boosting, stacking.
Non supervisé	Clustering	K-means.
	Réduction de dimension	Analyse en composantes principales (ACP).
	Détection d'anomalies	Isolation forest, One-Class SVM.
	Exploration d'associations	Apriori, FP-Growth.
Renforcement	Apprentissage par renforcement	Q-Learning.
Semi-supervisé	Classification et régression	Méthodes de propagation d'étiquettes.
Auto-apprentissage	Clustering et reconstruction	Autoencodeurs, réseaux de neurones générateurs (GAN).

TABLE 2.1 – Tableau des algorithmes d'apprentissage automatique par type d'apprentissage et tâches

Dans ce qui suit, nous définissons et expliquons quelques-uns de ces algorithmes.

2.4.1 Régression linéaire

La régression linéaire est une technique statistique utilisée pour modéliser et analyser les relations entre une variable dépendante et une ou plusieurs variables indépendantes. L'objectif est de trouver une fonction linéaire qui minimise les écarts entre les valeurs observées et les valeurs prédites [3]. La formule générale de la régression linéaire est :

$$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_m x_m \tag{2.5}$$

où :

- y est la variable dépendante,
- $x_1, x_2, \ldots, x_m$ sont les variables indépendantes,
- θ_0 est l'ordonnée à l'origine,
- $\theta_1, \theta_2, \ldots, \theta_m$ sont les coefficients de régression.

2.4.2 K-nearest neighbors (KNN)

L'algorithme des K plus proches voisins ou K−nearest neighbors (KNN) est un modèle d'apprentissage supervisé non paramétrique et non linéaire qui se base sur la proximité pour catégoriser ou prédire le regroupement d'un point de données spécifique [14].

Fonctionnement des KNN

l'algorithme consiste a étiqueter une nouvelle observation en fonction des étiquettes des k points les plus proches entourant un point de données cible. Dans un problème de classification, on utilise un vote majoritaire pour attribuer une étiquette à la nouvelle observation, celle-ci prenant l'étiquette majoritaire parmi celles de ses k plus proches voisins. Pour un problème de régression, la nouvelle observation est étiquetée en prenant comme valeur la moyenne des étiquettes de ses k plus proches voisins [14].

L'algorithme KNN fonctionne selon les étapes suivantes :

1. **Sélection des k voisins** :
   - Choisir la valeur de k (nombre de voisins).
2. **Calcul de la distance** :
   - Choisir une fonction de distance appropriée (distance euclidienne, distance de Manhattan).
   - Calculer la distance entre la nouvelle observation et chaque point de données existant dans le jeu de données.
3. **Sélection des k voisins les plus proches** :
   - Identifier les k points de données les plus proches de la nouvelle observation en fonction des distances calculées.

Formulation mathématique

Dans l'algorithme des k plus proches voisins (KNN), une fonction de calcul de distance entre deux observations est essentielle pour évaluer leur similarité, plus deux points sont proches, plus ils sont considérés comme similaires, et inversement. plusieurs fonctions de distance existent, chacune adaptée à des types de données spécifiques, comme la distance euclidienne pour des données quantitatives similaires et la distance de Manhattan pour des données de types différents. Le choix de la fonction de distance dépend des caractéristiques des données [3].

Voici comment KNN effectue ses prédictions :

- Pour un problème de classification
  - Calculer la probabilité d'appartenance de la nouvelle observation à chaque classe c, basée sur la fréquence des classes parmi les k plus proches voisins.
  - Attribuer à la nouvelle observation la classe avec la probabilité d'appartenance la plus élevée. l'étiquette prédite $f(x)$ est :

$$f(x) = \arg\max_{c} \sum_{i \in \mathcal{N}_k(x)} \mathbf{1}(y_i = c) \tag{2.6}$$

Dans cette équation :

- $f(x)$ représente la classe prédite pour un point de donnée x.
- $\arg\max_c$ indique que nous cherchons la classe c qui maximise la somme à l'intérieur.
- $\mathcal{N}_k(x)$ représente l'ensemble des k plus proches voisins.
- $\mathbf{1}(y_i = c)$ est la fonction indicatrice, qui vaut 1 si $y_i = c$ et 0 sinon.

- c parcourt l'ensemble des classes.

- Pour un problème de régression
  - Calculer la moyenne des valeurs des étiquettes des k voisins.
  - Attribuer cette moyenne comme valeur prédite pour la nouvelle observation.

  La valeur prédite $f(x)$ est :

$$f(x) = \frac{1}{k} \sum_{i \in \mathcal{N}_k(x)} y_i \tag{2.7}$$

Dans cette équation :

- $f(x)$ est la prédiction de sortie pour le point de données x.
- $\frac{1}{k}$ est le facteur de normalisation qui calcule la moyenne des valeurs de ces voisins.
- $\mathcal{N}_k(x)$ représente l'ensemble des k plus proches voisins.

La figure 2.8 représente un exemple d'un modèle KNN.

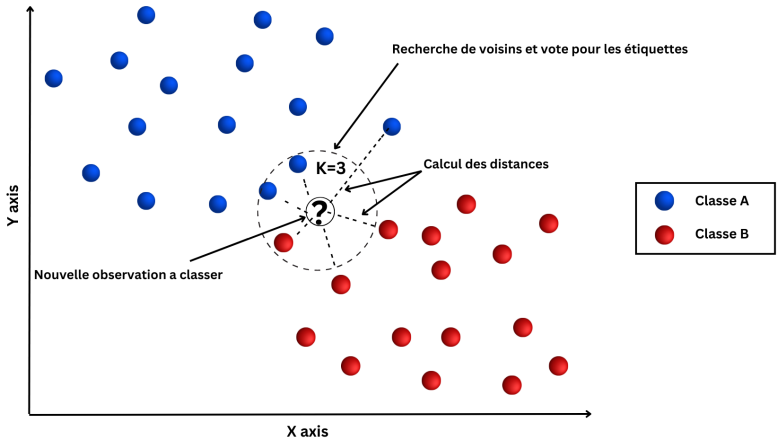

FIGURE 2.8 – Exemple d'un KNN

2.4.3 Arbre de décision

Un arbre de décision (decision tree) est une structure arborescente, proposant une diversité de choix potentiels, où chaque branche conduit à un résultat spécifique. Il s'agit d'un type d'algorithme d'apprentissage supervisé non paramétrique largement employé dans les tâches de classification et de régression en raison de leur interprétabilité, leur intégration aux bases de données, leur fiabilité et de leur simplicité algorithmique [6].

Fonctionnement des arbres de décision

Les arbres de décision se divisent en trois structures : les noeuds de décision, les branches et les feuilles [6].

- **Nœud racine** : Point d'accès principal à l'arbre.
- **Nœud interne** : Points de décision avec des descendants.
- **Nœuds terminaux (feuilles)** : Ces nœuds représentent des fins de branches et ne possèdent pas de descendants, ce qui signifie qu'ils n'ont pas de branche sortante.
- **Branche** : Elle détermine le résultat d'un test effectué sur les nœuds internes.

Chaque nœud interne teste une variable d'apprentissage, chaque branche en donne le résultat, et chaque feuille contient la valeur cible, soit une étiquette de classe pour la classification, soit une valeur numérique pour la régression. L'algorithme segmente les éléments en sous-groupes similaires pour construire un modèle prédictif basé sur les règles de décision.

Formulation mathématique

L'indice de Gini et l'entropie croisée sont des métriques couramment utilisées pour évaluer l'impureté des données lors de la construction de l'arbre de décision [6].

Indice de Gini : évalue la probabilité qu'un élément choisi aléatoirement dans un sous-ensemble soit mal classé. Son objectif est de minimiser cet indice pour garantir des sous-groupes homogènes. La valeur de l'indice de Gini varie entre 0 et 1, similaire à l'entropie. Sa formule de calcul est donnée par :

$$I_G = 1 - \sum_{i}^{n} (p_i)^2 \tag{2.8}$$

Entropie croisée ou déviance : mesure l'incertitude dans les données et est utilisée pour minimiser l'entropie lors de la construction de l'arbre. Sa formule de calcul est définie comme suit :

$$E_c = -\sum_{i}^{n} p_i \log_2(p_i) \tag{2.9}$$

La figure 2.9 illustre un arbre de décision.

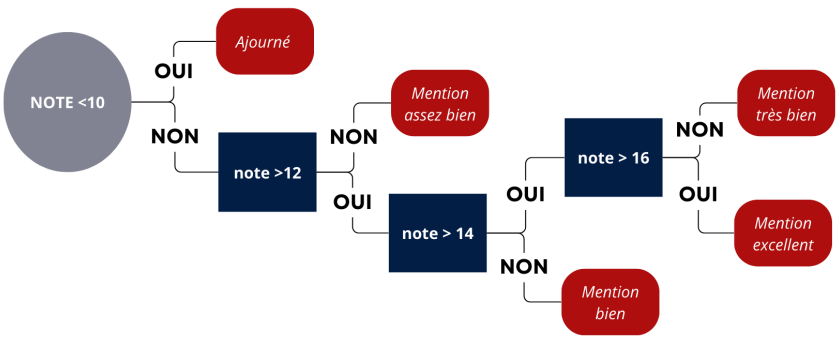

FIGURE 2.9 – Exemple d'un arbre de décision

Optimisation des hyperparamètres

Les principaux paramètres à optimiser pour l'algorithme Decision Tree Regressor sont :

- `max_depth` : La profondeur maximale de l'arbre. Un arbre plus profond peut modéliser des relations plus complexes mais risque de surapprendre les données d'entraînement.
- `min_samples_split` : Le nombre minimum d'échantillons requis pour diviser un nœud. Une valeur plus élevée empêche l'arbre de modéliser des relations trop spécifiques aux données d'entraînement.
- `min_samples_leaf` : Le nombre minimum d'échantillons requis dans une feuille. Une valeur plus élevée crée des feuilles contenant plus d'échantillons, ce qui peut également réduire le surapprentissage.

2.4.4 Méthodes d'ensemble séquentielles (boosting)

Le boosting est une technique d'apprentissage supervisé qui combine plusieurs modèles faibles pour créer un modèle puissant. Un modèle faible est légèrement meilleur que le hasard, mais en les combinant de manière itérative, on obtient une amélioration significative des performances. Les trois algorithmes de boosting les plus populaires sont AdaBoost, Gradient Boosting et XGBoost.

AdaBoost (adaptive boosting)

AdaBoost, introduit par Freund et Schapire en 1997, est une méthode d'apprentissage ensembliste largement utilisée qui cherche à agréger plusieurs classificateurs (généralement des arbres de décision) de faible profondeur appelés *stumps* afin de former un classificateur robuste et précis. Il fonctionne en ajustant les poids des observations, en mettant davantage l'accent sur les observations mal classées par les modèles précédents [11].

Fonctionnement de AdaBoost

Le fonctionnement de AdaBoost repose sur une approche itérative où chaque modèle faible est formé sur une version pondérée de l'ensemble de données. Les poids des instances mal classées sont augmentés, tandis que ceux des instances correctement classées sont diminués, forçant les modèles successifs à se concentrer sur les exemples difficiles [11].

Le fonctionnement de l'algorithme AdaBoost repose sur une approche itérative avec les étapes suivantes :

1. **Initialisation des poids** :
   - Initialiser les poids de toutes les observations de l'ensemble de données de manière uniforme. Si le nombre total d'observations est n, chaque observation reçoit un poids initial.
2. **Entraînement des classificateurs faibles** :
   - Pour chaque itération t (de 1 à T, où T est le nombre total de classificateurs faibles) :
     - Entraîner un classificateur faible sur l'ensemble de données pondéré.

- Calculer l'erreur ϵ_t du classificateur en fonction des poids des observations.

3. **Calcul de la pondération du classificateur** :
   - Calculer la pondération du classificateur en fonction de son erreur ϵ_t.
4. **Mise à jour des poids des observations** :
   - Mettre à jour les poids des observations en augmentant les poids des observations mal classées et en diminuant les poids des observations correctement classées.
   - Normaliser les poids pour qu'ils forment une distribution de probabilité.

Ces étapes permettent à AdaBoost de se concentrer sur les exemples difficiles à chaque itération, améliorant ainsi progressivement la précision globale du modèle [11]. La figure 2.10 représente l'entraînement du modèle AdaBoost.

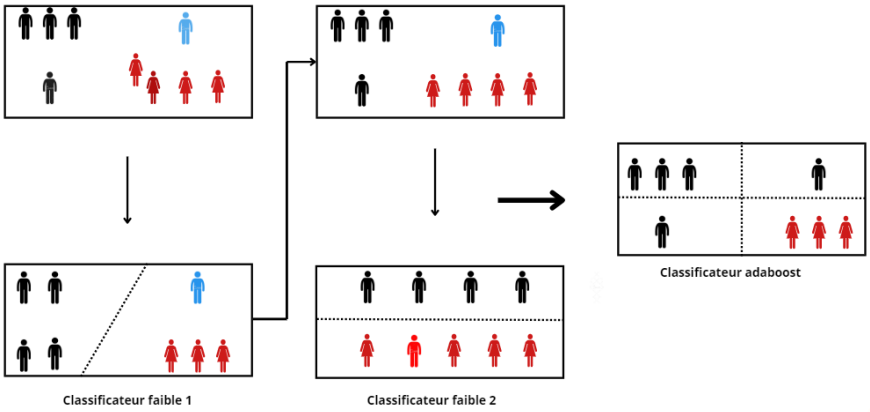

FIGURE 2.10 – Entraînement d'AdaBoost

XGBoost (extreme gradient boosting)

XGBoost développé par Tianqi Chen et initialement publié en 2014 [7], est une implémentation optimisée de l'algorithme de boosting Decision Tree qui se distingue par son efficacité dans les tâches de régression et de classification.

Reconnu pour sa robustesse et sa performance, il est particulièrement adapté aux défis posés par la prévision de données de vente, qui peuvent être complexes et influencées par de multiples variables.

Fonctionnement de XGBoost

L'algorithme repose sur un processus d'apprentissage itératif optimisé construisant un ensemble de modèles faibles (typiquement des arbres de décision). A chaque itération, un nouveau modèle est créé pour corriger les erreurs des modèles précédents.

Les prédictions de ce nouveau modèle sont ensuite intégrées à l'ensemble, contribuant ainsi à l'amélioration progressive de la précision globale.

Fonctionnement de XGBoost

L'algorithme repose sur un processus d'apprentissage itératif construisant un ensemble de modèles faibles (typiquement des arbres de décision). À chaque itération, un nouveau modèle est créé pour corriger les erreurs des modèles précédents. Les prédictions de ce nouveau modèle sont ensuite intégrées à l'ensemble, contribuant ainsi à l'amélioration progressive de la précision globale.

XGBoost améliore ce processus grâce à des optimisations telles que :

1. **Initialisation** :
   - Démarrer avec un modèle initial, souvent un arbre de décision simple, pour faire les premières prédictions $F_0(x)$.
2. **Calcul des résidus** :
   - Calculer les résidus (erreurs) entre les prédictions actuelles et les valeurs réelles des données d'entraînement.

$$g_i = \left. \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)} \right|_{F(x)=F_{t-1}(x)} \tag{2.10}$$

   et

$$h_i = \left. \frac{\partial^2 L(y_i, F(x_i))}{\partial F(x_i)^2} \right|_{F(x)=F_{t-1}(x)} \tag{2.11}$$

3. **Entraînement du nouveau modèle** :
   - Entraîner un nouvel arbre de décision sur les résidus calculés.

$$\sum_{i=1}^{N} \left(\frac{1}{2} h_i h_t(x_i)^2 + g_i h_t(x_i) \right) + \Omega(h_t) \tag{2.12}$$

   où $\Omega(h_t)$ est un terme de régularisation.
4. **Mise à jour des prédictions** :
   - Mettre à jour les prédictions en ajoutant les prédictions du nouvel arbre au modèle existant.

$$F_t(x) = F_{t-1}(x) + \nu h_t(x) \tag{2.13}$$

5. **Optimisations de XGBoost** :
   - **Régularisation** : Intégrer une régularisation explicite pour éviter le surapprentissage.
   - **Gestion des valeurs manquantes** : Traiter efficacement les valeurs manquantes en apprenant la meilleure direction à prendre lors de la division des arbres.
   - **Régularisation L1 (lasso) et L2 (ridge)** : Ajouter une pénalité à la complexité du modèle pour éviter le surapprentissage.
   - **Parallélisme** : Utiliser le parallélisme pour accélérer les calculs pendant la formation du modèle.

- **Pruning** : Appliquer un pruning (élagage) post-ordre pour supprimer les branches inutiles et réduire le surapprentissage.

6. **Répétition** :
   - Répéter les étapes de calcul des résidus, d'entraînement du nouvel arbre de décision, et de mise à jour des prédictions jusqu'à ce qu'un nombre prédéterminé d'arbres soit atteint ou que l'amélioration de la performance devienne négligeable.

Ces étapes permettent à XGBoost de construire un modèle robuste et précis, capable de gérer efficacement des jeux de données complexes et variés [7].

Formulation mathématique

Le modèle final est donné par :

$$F_T(x) = F_0(x) + \sum_{t=1}^{T} \nu h_t(x) \tag{2.14}$$

Optimisation des hyperparamètres

Les principaux paramètres à optimiser pour le modèle XGBoost sont :

- `n_estimators` : Le nombre d'arbres à utiliser dans le modèle. Un nombre plus élevé peut améliorer la performance mais augmente le risque de surapprentissage.
- `learning_rate` : Le taux d'apprentissage. Un taux plus faible peut améliorer la performance du modèle mais nécessite plus d'arbres.
- `max_depth` : La profondeur maximale des arbres. Une profondeur plus élevée permet de modéliser des relations plus complexes.
- `gamma` : Le seuil minimal de perte de répartition nécessaire pour effectuer une partition supplémentaire.
- `subsample` : La proportion des échantillons utilisés pour chaque arbre. Une valeur plus faible réduit le surapprentissage.
- `colsample_bytree` : La proportion des caractéristiques utilisées pour chaque arbre. Une valeur plus faible réduit le surapprentissage.

CatBoost (categorical boosting)

CatBoost développé par Yandex en 2017 a été conçu afin d'améliorer les performances du gradient boosting pour gérer efficacement les variables catégorielles et réduire le surapprentissage, offrant ainsi des performances robustes et précises sur des ensembles de données hétérogènes [16].

Fonctionnement de CatBoost

CatBoost crée de manière itérative une série de modèles faibles (arbres de décision). Chaque modèle suivant corrige les erreurs du modèle précédent [16].

1. **Prétraitement des données** :
   - Encodage des variables catégorielles en utilisant une technique appelée "cible statistique" ou "ciblage par moyennes", qui remplace les valeurs catégorielles par une transformation basée sur la cible et les statistiques des catégories.

$$\text{Valeur encodée } (x_i) = \frac{\sum_{j \neq i} y_j}{N - 1} \tag{2.15}$$

     où x_i est une valeur catégorielle, y_j est la cible, et N est le nombre total d'observations.
   - Gestion des valeurs manquantes en les traitant comme des catégories distinctes.
2. **Initialisation des poids** :
   - Initialiser les poids des observations de l'ensemble de données.
3. **Construction des arbres de décision** :
   - Pour chaque itération t (de 1 à T, où T est le nombre total de modèles faibles) :

$$\mathrm{r}_i^{(t)} = y_i - \hat{y}_i^{(t-1)} \tag{2.16}$$

     où y_i est la valeur cible réelle et $\hat{y}_i^{(t-1)}$ est la prédiction à l'itération précédente.
   - Entraîner un arbre de décision sur l'ensemble de données pondéré.
   - Utiliser le gradient des erreurs résiduelles des prédictions actuelles pour guider la croissance de l'arbre.
4. **Mise à jour des prédictions** :
   - Mettre à jour les prédictions en ajoutant les prédictions du nouvel arbre au modèle existant.

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + \eta \cdot f_t(x_i) \tag{2.17}$$

     où η est le taux d'apprentissage et f_t est le nouvel arbre de décision.
5. **Régularisation et contrôle du surapprentissage** :
   - Utiliser des techniques de régularisation pour contrôler la complexité du modèle, y compris des pénalités L2 et des techniques de bagging.

$$\text{Terme de régularisation} = \lambda \sum_{j=1}^{n} w_j^2 \tag{2.18}$$

     où λ est le paramètre de régularisation, w_j représente les poids du modèle, et n est le nombre total de poids.
   - Appliquer un élagage post-ordre pour supprimer les branches inutiles des arbres.
6. **Combinaison des modèles** :
   - Combiner les modèles faibles de manière pondérée pour obtenir le modèle final.

Ces étapes permettent à CatBoost de gérer efficacement les variables catégorielles et de construire des modèles robustes et précis, adaptés aux ensembles de données hétérogènes [16].

Formulation mathématique

CatBoost se distingue par l'utilisation de permutations de données pour gérer efficacement les variables catégorielles, ce qui réduit le biais et améliore la généralisation [16].
De plus, ces techniques avancées permettent de minimiser le risque de surapprentissage par rapport aux autres méthodes de boosting [16].

Le modèle final est une combinaison pondérée de tous les modèles faibles :

$$H(x) = \sum_{t=1}^{T} \alpha_t \cdot f_t(x) \tag{2.19}$$

où α_t est le poids du modèle t et $f_t(x)$ est le modèle faible à l'itération t.

Optimisation des hyperparamètres

Les principaux paramètres à optimiser pour l'algorithme `CatBoost` sont :

- `itérations` : Le nombre d'arbres à construire. Un nombre plus élevé peut améliorer la performance mais augmente le risque de surapprentissage.
- `learning_rate` : Le taux d'apprentissage. Un taux plus faible peut améliorer la performance du modèle mais nécessite plus d'arbres.
- `depth` : La profondeur des arbres. Une profondeur plus élevée permet de modéliser des relations plus complexes.
- `l2_leaf_reg` : Le coefficient de régularisation L2. Une régularisation plus forte peut aider à prévenir le surapprentissage.
- `border_count` : Le nombre de seuils pour les caractéristiques numériques. Plus de seuils permettent une modélisation plus fine mais peuvent augmenter le temps de calcul.

2.4.5 Méthodes d'ensemble parallèles (bagging)

Le bagging, est une méthode d'ensemble qui améliore la précision et la stabilité des algorithmes de ML. Il fonctionne en créant plusieurs versions du modèle d'origine à partir de sous-échantillons des données de formation et en agrégeant les prédictions de ces modèles.

Forêts aléatoires

Random forest (forêt aléatoire) est un modèle d'apprentissage supervisé utilisé en classification et en régression. Il s'appuie sur la méthode du bagging [4].

Fonctionnement de forêt aléatoire

Chaque arbre est construit avec une sélection aléatoire de variables, ce qui augmente la diversité des arbres. En agrégeant les prédictions de chaque arbre, la forêt aléatoire fournit une prédiction finale robuste, réduisant le risque de sur-apprentissage. Ce modèle est efficace pour gérer des données complexes et bruitées, et est résilient aux valeurs aberrantes [4].

1. **Construction des arbres :**
   - Un nombre fixe d'arbres n est défini.
   - Pour chaque arbre i :
     - Un échantillon bootstrap D_i est tiré aléatoirement avec remplacement à partir de l'ensemble de données D.
     - Un arbre de décision T_i est construit en utilisant cet échantillon D_i. À chaque nœud de l'arbre, une sous-sélection de m caractéristiques est choisie aléatoirement parmi les p caractéristiques disponibles. La meilleure division est choisie parmi ces m caractéristiques.
2. **Prédiction :**
   Pour une nouvelle observation x :
   - Chaque arbre T_i fournit une prédiction y_i.
   - Pour la classification, la prédiction finale est obtenue par vote majoritaire parmi les prédictions y_i.
   - Pour la régression, la prédiction finale est la moyenne des prédictions y_i.

Formulation mathématique

Soit $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ l'ensemble de données d'entraînement où x_i représente une caractéristique et y_i la variable cible [4].

1. **Échantillonnage Bootstrap :**
   Pour chaque arbre T_i, un échantillon bootstrap D_i est généré à partir de D.

2. **Sélection des caractéristiques :**
   - À chaque nœud de l'arbre, une sous-sélection m des p caractéristiques est faite.
   - La division est choisie parmi les m caractéristiques en utilisant une mesure comme le critère Gini pour la classification ou la variance réduite pour la régression.

3. **Construction de l'arbre :**
   L'arbre est construit en utilisant l'algorithme CART (Classification and Regression Trees) sur l'échantillon bootstrap D_i avec la restriction de m caractéristiques à chaque division.

4. **Prédiction finale :**
   Pour une nouvelle observation x :
   - **Classification** :

$$f(x) = \arg\max_c \left(\frac{1}{B} \sum_{b=1}^{B} 1(T_b(x) = c) \right) \tag{2.20}$$

     - $f(x)$ représente la prédiction de classe pour l'observation x.

- $\arg\max_c$ sélectionne la classe c qui maximise l'expression qui suit.
- $\frac{1}{B}$ est le facteur de normalisation pour calculer la moyenne sur tous les arbres B.
- $\sum_{b=1}^{B}$ indique la somme sur tous les arbres de la forêt B.
- $1(T_b(x) = c)$ est la fonction indicatrice qui vaut 1 si l'arbre b prédit la classe c pour x, et 0 sinon.

- **Régression** :

$$\widehat{y} = \frac{1}{n} \sum_{i=1}^{n} y_i \tag{2.21}$$

- $\hat{y}$ représente la prédiction de la variable y.
- n est le nombre total d'observations.
- y_i sont les valeurs observées pour chaque observation i.
- $\sum_{i=1}^{n} y_i$ est la somme des valeurs observées sur l'ensemble des n observations.

Optimisation des hyperparamètres

Ajuster les paramètres tels que le nombre d'arbres et de variables permet d'optimiser les performances du modèle. Cette approche mathématique garantit des prédictions précises et généralisables.

Les principaux paramètres à optimiser pour l'algorithme Random forest sont :

- `n_estimators` : Le nombre d'arbres dans la forêt. Un nombre plus élevé peut améliorer la performance mais augmente le temps de calcul.
- `max_depth` : La profondeur maximale des arbres. Une profondeur plus élevée permet de modéliser des relations plus complexes.
- `min_samples_split` : Le nombre minimum d'échantillons requis pour diviser un nœud. Une valeur plus élevée empêche l'arbre de modéliser des relations trop spécifiques aux données d'entraînement.
- `min_samples_leaf` : Le nombre minimum d'échantillons requis dans une feuille. Une valeur plus élevée crée des feuilles contenant plus d'échantillons, ce qui peut réduire le surapprentissage.
- `max_features` : Le nombre de caractéristiques à considérer pour trouver la meilleure division. `auto` utilise toutes les caractéristiques, `sqrt` utilise la racine carrée du nombre de caractéristiques, et `log2` utilise le logarithme en base 2 du nombre de caractéristiques.

2.4.6 Stacking

Le stacking est une technique sophistiquée d'apprentissage automatique qui vise à améliorer les performances de prédiction en combinant les résultats de différents modèles de base. Contrairement aux méthodes plus simples comme l'agrégation par moyennes ou par votes majoritaires, le stacking utilise un méta-modèle pour apprendre à partir des prédictions des modèles de base.

Dans le processus de stacking, plusieurs modèles de base (régressions, arbres de décision, réseaux de neurones) sont d'abord entraînés indépendamment sur les mêmes données d'entraînement. Ensuite, les prédictions générées par ces modèles de base servent d'entrées à un méta-modèle, souvent un modèle plus simple comme une régression linéaire ou une régression logistique. Le méta-modèle est alors entraîné à prédire la sortie finale à partir de ces prédictions de base.

L'avantage principal du stacking réside dans sa capacité à capturer les différentes perspectives des modèles de base, réduisant ainsi le biais et la variance des prédictions finales. La figure 2.11 illustre le processus de stacking, où les prédictions des modèles de base sont combinées par le méta-modèle pour produire une prédiction finale plus précise [14].

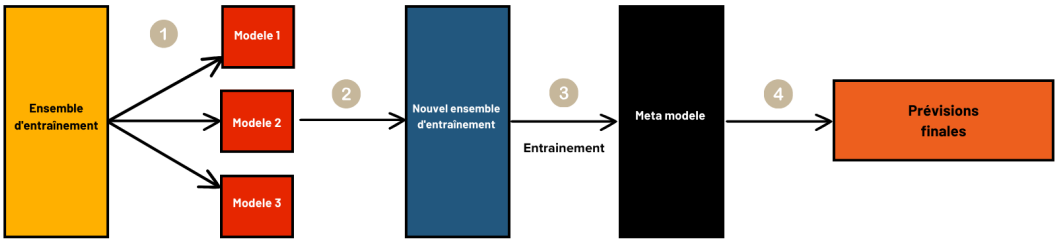

FIGURE 2.11 – Processus d'assemblage

2.5 Apprentissage profond

L'apprentissage profond, une branche de l'apprentissage automatique, utilise des réseaux de neurones artificiels profonds pour extraire automatiquement des caractéristiques complexes des données brutes, contrairement aux algorithmes traditionnels nécessitant une ingénierie manuelle des caractéristiques. Cette approche hiérarchique est efficace dans divers domaines, tels que la reconnaissance d'images et la compréhension du langage naturel. Cependant, le deep learning requiert souvent de grandes quantités de données et des ressources de calcul importantes, ce qui peut limiter son utilisation dans certains contextes [20].

2.6 Optimisation et validation des modèles

Cette section se concentre sur les techniques d'optimisation et de validation des modèles d'apprentissage automatique. Nous examinerons les méthodes utilisées pour améliorer la performance des modèles, ainsi que les approches permettant de garantir leur robustesse et leur généralisation sur des données non vues [15].

Validation croisée

Pour la validation et le bon choix des paramètres, nous avons opté pour la validation croisée. Cette dernière, est une technique utilisée pour évaluer la capacité de généralisation d'un modèle. Elle consiste à diviser les données en plusieurs sous-ensembles (ou folds) et à entraîner le modèle sur certains de ces sous-ensembles tout en le testant sur les autres. Cela permet d'obtenir une estimation plus robuste de la performance du modèle.

Pour les séries chronologiques, la validation croisée traditionnelle n'est pas appropriée car elle ne respecte pas l'ordre chronologique des données.
Nous avons donc utilisé `TimeSeriesSplit`, qui est une méthode de validation croisée spécifiquement conçue pour les séries chrnologiques [15].

GridSearchCV

GridSearchCV est une méthode utilisée pour automatiser la recherche des meilleurs hyperparamètres pour un modèle donné en effectuant une validation croisée sur une grille de paramètres spécifiée. Elle permet d'optimiser les performances du modèle en testant de manière exhaustive différentes combinaisons de paramètres.

Cette méthode effectue une recherche exhaustive parmi une grille spécifiée de paramètres et sélectionne la combinaison qui minimise l'erreur quadratique moyenne négative (neg_mean_squared_error) à l'aide de la validation croisée en série temporelle (TimeSeriesSplit).

Les principaux paramètres utilisés dans GridSearchCV sont [15] :

- `estimator` : L'algorithme de machine learning à utiliser.
- `param_grid` : Le dictionnaire de la grille des paramètres à tester.
- `cv` : La stratégie de validation croisée (ici, TimeSeriesSplit est utilisé pour les séries temporelles).
- `scoring` : La métrique utilisée pour évaluer les performances du modèle.
- `n_jobs` : Le nombre de tâches à exécuter en parallèle.

2.7 Métriques d'évaluation des modèles

Les métriques d'évaluation utilisées pour les problèmes de régression sont [24] :

- Erreur absolue moyenne (Mean Absolute Error - MAE) : L'erreur absolue moyenne représente la moyenne des écarts entre les valeurs prédites et les valeurs réelles. Elle quantifie la distance entre les prévisions et les sorties réelles. Elle est définie comme :

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{2.22}$$

  Où Y_i sont les valeurs observées, $\widehat{y_i}$ sont les valeurs prédites, et n est le nombre d'observations.

- Erreur quadratique moyenne (Mean Squared Error - MSE) : Le MSE calcule la moyenne des carrés des écarts entre les valeurs prédites et les valeurs réelles. Cela permet de mettre davantage l'accent sur les erreurs importantes. Sa formule est donnée comme suit :

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2 \tag{2.23}$$

- Erreur quadratique moyenne racine (Root Mean Squared Error - RMSE) : Celle-ci est la racine carrée du MSE et est une métrique couramment utilisée pour évaluer la précision des prédictions numériques.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{2.24}$$

- Coefficient de détermination R^2 : mesure la proportion de la variance totale des données qui est expliquée par le modèle de régression. Il est calculé à partir de la formule suivante :

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{2.25}$$

2.8 Travaux connexes

Dans le cadre de notre étude, nous avons examiné plusieurs travaux portant sur la prédictions, qui ont utilisé diverses méthodes. Parmi ces recherches, nous avons retenu les suivantes :

2.8.1 Apprentissage automatique supervisé

En 2016, Tack et al. [22] ont exploré l'application de méthodes d'apprentissage supervisé incrémental permettant une mise à jour continue des paramètres du modèle à mesure que de nouvelles données sont disponibles, afin d'estimer le niveau de compétence lexicale chez les apprenants du français langue étrangère (FLE).
Lors de leur étude présentée en conférence *conjointe JEP-TALN-RECITAL*, ils ont proposé deux approches : un modèle basé sur la compétence lexicale moyenne des apprenants du même niveau et un modèle personnalisé pour chaque apprenant. Les résultats montrent que les modèles personnalisés (M_P) sont significativement plus précis que les modèles basés sur la moyenne du niveau (M_L) et le modèle de référence (M_E). En particulier, les réseaux de neurones artificiels ont démontré une robustesse supérieure par rapport aux SVM, avec une amélioration de l'exactitude allant de 9% à 17%. De plus, le sous-échantillonnage de la classe majoritaire de mots connus a amélioré la prédiction. Les modèles personnalisés ont montré une précision significativement supérieure par rapport au modèle de référence, soulignant les avantages de l'approche individualisée pour la prédiction des compétences lexicales.

En 2018, Les travaux de Cheriyan et al. [8] se sont concentrés sur la prédiction des ventes en utilisant des techniques de machine learning, tels que le modèle linéaire généralisé (GLM), l'arbre de décision (DT) et l'arbre de gradient boosté (GBT), pour la prédiction des tendances de ventes. Leur étude présentée lors de la conférence *iCCECE* et basée sur les données de ventes

d'un magasin de mode en ligne sur trois ans (2015-2017) comprenant la catégorie, la ville, la description d'articles, la quantité, la semaine, le trimestre, l'année, le chiffre d'affaires, la description de l'UGS, le type et le nombre, démontre que l'algorithme de GBT offre la meilleure performance avec une précision de 98 %, surpassant les modèles DT (71%) et GLM (64%). Les performances ont été évaluées en termes de précision de classification, taux d'erreur, précision, rappel et coefficient de Kappa. Les prévisions de ventes futures, de 2018 à 2021, ont montré une augmentation légère des ventes, avec une analyse des tendances saisonnières indiquant des variations importantes en janvier et août. Le défi majeur rencontré était la gestion d'un ensemble volumineux de données, comprenant environ 85 000 enregistrements, pour la comparaison des algorithmes de prévision. La complexité et le temps d'exécution élevé ont conduit à la suppression de certains enregistrements. De plus, l'insuffisance des champs et attributs a limité la profondeur des analyses. Les auteurs ont conclu que l'utilisation de systèmes de prévision d'apprentissage automatique, notamment le modèle GBT, qui s'est avéré très performant, peut améliorer considérablement la précision des prévisions de ventes et des décisions commerciales grâce à sa capacité à traiter de grandes quantités de données.

En 2021, Ulrich et al. [23] ont proposé une nouvelle approche consistant à classer les motifs de demande pour chaque produit, puis à choisir le modèle de prévision de demande le mieux adapté à chaque catégorie. Leur étude, publiée dans *l'International Journal of Forecasting*, suggère que cette méthode de segmentation pourrait grandement améliorer la précision des prévisions de vente au détail. Ils ont conçu un système automatisé qui traite la sélection du modèle de prévision comme un problème de classification. Entraîné sur des données historiques provenant d'un détaillant en ligne de produits d'épicerie, couvrant trois ans et 111 SKU de cinq centres de distribution différents, l'algorithme apprend à associer les profils de demande aux modèles les plus performants. Il peut ainsi recommander le plus optimal pour de nouvelles données, en fonction des caractéristiques observées. Les chercheurs ont comparé les performances de multiples techniques de prévision, allant de la régression linéaire à des modèles avancés comme les GAMLSS, la régression quantile, les forêts de régression quantile et ARIMAX. Leurs résultats ont montré que la performance des modèles varie selon les produits et les périodes de demande, soulignant ainsi la flexibilité nécessaire des solutions sur mesure. Ils ont constaté qu'aucun modèle ne surpasse systématiquement les autres, mettant en avant la nécessité d'une approche adaptable.

Pour optimiser cette sélection, ils ont employé un méta-modèle, qui comprend un ensemble de caractéristiques des données et des prévisionnistes de base, ainsi que des connaissances acquises sur la méthode de prévision la plus appropriée en fonction de ces caractéristiques. Ce méta-modèle agit comme un algorithme de classification, apprenant à partir des données historiques comment associer les motifs de demande aux modèles de prévision les plus performants.

En 2023, les travaux de Agomoh et Ukabuiro [1] publiés dans *l'European Journal of Theoretical and Applied Sciences*, se sont concentrés sur la conception et la mise en œuvre d'un système de prévision des ventes basé sur le ML pour améliorer la capacité de production et la trajectoire des ventes d'une entreprise de brasserie théorique, désignée comme *Company*.

Ils ont développé un modèle de régression linéaire multiple, particulièrement adapté à l'analyse des tendances, et l'ont déployé sur un serveur web local pour permettre aux utilisateurs de générer des prévisions via une interface web. Le modèle a été entraîné et testé avec des données historiques en exculant les données de 2020 en raison de leur écart significatif par rapport aux autres années, probablement dû à des événements externes influençant les ventes. Les perfor-

mances ont été évaluées à l'aide de l'erreur quadratique moyenne RMSE et de l'erreur absolue moyenne MAE, obtenant des scores respectifs de 2,36 et 1,76.
Ces résultats montrent l'efficacité du modèle pour prédire les ventes avec précision. Les auteurs concluent que l'application de techniques de ML peut significativement améliorer la précision des prévisions de ventes, et recommandent l'utilisation de ces approches pour optimiser la planification stratégique et la prise de décision en entreprise.

2.8.2 Apprentissage profond

En 2019, Yu-Sen Shih et Min-Huei Lin [21] ont proposé un modèle combinant les réseaux de neurones LSTM et l'analyse de sentiments des commentaires clients pour améliorer les prévisions de ventes à court terme dans le e-commerce. Cette approche hybride a été appliquée aux données de vente de *Gift box multi-taste Gift chocolate* pendant une période de 36 jours, les données de ventes ont été collectées via une extension Chrome, tandis que les commentaires ont été extraits à l'aide de l'API ouverte de *taobao.com*. Un total de 5018 commentaires a été rassemblé, incluant des informations telles que l'identifiant du membre, la date du commentaire et le contenu complet des commentaires et ont été organisés par date de commentaire, et tous les commentaires d'un même jour ont été concaténés en un seul article. Ces articles ont ensuite été joints aux données de ventes par date de commentaire. L'analyse des sentiments a été réalisée en utilisant l'API d'analyse des sentiments de *ai.baidu.com*, permettant de classer les sentiments en "positif", "négatif" et "confiance". Le modèle utilisé pour la prévision des ventes est un réseau neuronal récurrent LSTM. Pendant la phase d'entraînement, les données de séries temporelles normalisées, incluant les ventes et les caractéristiques des sentiments, sont utilisées pour prédire les ventes futures. La fonction de perte utilisée est la MSE et l'optimiseur est Adam. .Présentée lors de la conférence *ACIIDS*, elle vise à exploiter les données textuelles en plus des données de ventes pour gagner en précision prédictive.

Les résultats démontrent que fusionner analyse sémantique et modèles neuronaux accroît la justesse prédictive pour prévoir les ventes produits court terme. Comparant les MAPE avec ou sans composante textuelle, les algorithmes exploitant les avis clients s'avèrent supérieurs. Toutefois, pondérer différemment ces données expressives optimise parfois davantage la précision.

En 2022, la thèse de Rémy Garnier [12] intitulée *Machine Learning sur les séries temporelles et applications à la prévision des ventes pour l'e-Commerce*, a exploré en profondeur l'application des modèles de l'apprentissage automatique aux séries temporelles dépendantes. Garnier a proposé deux modèles principaux : ARBoost, un modèle régressif basé sur des algorithmes de boosting intégrant la saisonnalité et des variables externes, et un modèle de compétitivité conçu pour traiter la cannibalisation des ventes entre produits concurrents en utilisant des réseaux de neurones. Appliqués aux données de CDiscount, ces modèles ont montré une précision accrue dans les prévisions de ventes, évitant notamment la sous-prédiction classique. Cette recherche souligne l'importance des techniques avancées du ML pour améliorer la précision des prévisions de ventes et propose des solutions pratiques pour des décisions commerciales plus informées et stratégiques. Les principales métriques d'évaluation utilisées pour mesurer la performance des modèles sont le RMSE, le MAE et le %MAE. Le RMSE optimise la moyenne des erreurs quadratiques, ce qui peut être sensible aux anomalies, tandis que le MAE optimise la médiane des erreurs absolues, souvent sous-estimant légèrement les ventes. Pour mieux modéliser la dispersion des données, une métrique de perte Poisson a également été introduite, en pénalisant plus fortement la sous-prédiction pour contrer la tendance naturelle des algorithmes

de ML à sous-prédire les ventes. Les résultats obtenus par Garnier sont significatifs : pour le modèle ARBoost, le RMSE a été réduit de 15% par rapport aux modèles classiques, tandis que le MAE a montré une amélioration de 10%. Le %MAE a également été réduit, indiquant une performance globale accrue du modèle de boosting. Le modèle de compétitivité a quant à lui démontré une amélioration des prévisions des parts de marché des produits, offrant une vision plus fine de la répartition des ventes entre produits concurrents, ce qui est essentiel pour les stratégies de tarification et de promotion. Le modèle de compétitivité proposé par Garnier modélise les ventes des produits d'une même catégorie en tenant compte de la cannibalisation des ventes entre produits concurrents. Ce modèle hiérarchique distribue la valeur agrégée des ventes en fonction des paramètres de compétitivité des produits, ce qui permet d'expliquer les variations des parts de marché à l'aide de covariables telles que le prix et la marge. En résumé, la thèse de Garnier démontre comment des techniques avancées de machine learning peuvent améliorer significativement les prévisions de ventes pour le e-commerce, en utilisant des modèles adaptés et des métriques d'évaluation robustes.

En 2023, Ahmadov et Helo [2] ont mené une recherche publiée dans *Discover Artificial Intelligence*. Leur étude vise à modéliser et à prévoir la demande de ventes en ligne intermittentes en utilisant des données provenant d'*eBay* et *GittiGidiyor* sur une période de deux à quatre ans. Leur recherche explorait l'efficacité des réseaux neuronaux profonds (DNNs) pour anticiper les fluctuations de ventes, comparant ces méthodes aux approches classiques telles que les moyennes mobiles, les lissages exponentiels, Croston et ARIMA. Les résultats principaux ont démontré une amélioration significative de la précision : les réseaux neuronaux profonds ont surpassé les modèles classiques jusqu'à 35%. De plus, l'utilisation de la distribution exponentielle de Poisson a permis de générer des prévisions proches des ventes réelles avec une marge d'erreur inférieure à 7%. Les analyses ont également révélé que les intervalles entre les commandes suivaient une distribution exponentielle, tout comme les tailles de commandes. En utilisant des données provenant de près de 3000 commandes de 17 vendeurs différents, l'étude a mis en évidence la performance supérieure des modèles de deep learning par rapport aux approches classiques, en particulier pour la précision prédictive.

Étude comparative des approches de prédiction des ventes

L'étude comparative des approches de prédiction des ventes met en lumière les différences significatives entre diverses méthodes d'apprentissage automatique et profond, chacune ayant ses avantages et ses limitations dans des contextes spécifiques.

Tack et al. (2016) [22] ont mis en avant l'efficacité des modèles personnalisés basés sur l'apprentissage supervisé incrémental, qui ont montré une amélioration significative de la précision par rapport aux méthodes classiques telles que les SVM. En comparaison, Cheriyan et al. (2018) [8] ont évalué plusieurs algorithmes, dont les GLM, DT et GBT, et ont trouvé que le GBT offrait la meilleure précision avec 98%, surpassant largement les autres méthodes testées. Yu-Sen Shih et Min-Huei Lin (2019) [21] ont introduit l'utilisation des LSTM combinés à une analyse de sentiments pour améliorer les prévisions à court terme des ventes, soulignant ainsi l'importance croissante des techniques modernes et de l'intégration de données textuelles, comme les commentaires clients, dans les modèles prédictifs. Ulrich et al. [23] (2021), quant à eux, ont adopté une approche plus diversifiée en testant la régression linéaire, les GAMLSS, la régression quantile et les forêts de régression quantile. Leur conclusion indique que la performance des modèles varie considérablement selon les caractéristiques spécifiques des produits et

des périodes de demande. ils ont obtenu des scores de RMSE et de MAE respectivement de 2,36 et 1,76, indiquant une haute précision dans leurs prévisions de ventes. Rémy Garnier en 2022 [12] a introduit l'ARBoost comme une méthode innovante qui intègre des éléments de compétitivité pour améliorer la précision des prévisions de ventes, soulignant ainsi l'importance de considérer les aspects concurrentiels dans les modèles de prédiction. D'un autre côté, Agomoh et Ukabuiro (2023) [1] ont validé l'efficacité de la régression linéaire multiple pour prédire les ventes, en montrant des scores de RMSE et MAE indiquant une bonne performance. Enfin, les travaux d'Ahmadov et Helo (2023) [2] tout comme ceux de Rémy Garnier, soulignent l'importance cruciale des techniques avancées de machine learning pour améliorer les prévisions de ventes, particulièrement dans le contexte dynamique du commerce électronique. Des modèles comme ARBoost et les réseaux neuronaux profonds offrent des solutions robustes, adaptées aux spécificités des données comme la saisonnalité et la cannibalisation des ventes.

En termes de critiques, Les modèles personnalisés offrent une précision améliorée mais sont complexes et coûteux en termes de calcul, nécessitant une infrastructure robuste pour les mises à jour continues. Les modèles GBT, bien que performants, consomment beaucoup de ressources computationnelles, entraînant des temps d'exécution élevés qui posent problème pour de grands ensembles de données ou des applications en temps réel. L'intégration de l'analyse des sentiments dans les modèles LSTM ajoute de la complexité, demandant des ressources supplémentaires pour le traitement des données textuelles. La méthode de segmentation des motifs de demande est complexe et nécessite beaucoup de données historiques. Les modèles ARBoost et de compétitivité sont efficaces mais complexes à paramétrer et optimiser, demandant des connaissances approfondies en apprentissage automatique. Les modèles de régression linéaire multiple, bien que simples et faciles à interpréter, manquent de précision dans des contextes de ventes dynamiques. Enfin, les réseaux neuronaux profonds, malgré leur haute performance, sont complexes et gourmands en ressources computationnelles. Le tableau 2.2 récapitule les travaux connexes précédemment cités.

Année	Auteurs	Approche	Résultats
2016	Tack et al. [22]	Apprentissage supervisé incrémental	Les modèles personnalisés (MP) sont plus précis que les modèles basés sur la moyenne du niveau (ML) et le modèle de référence (ME), améliorant l'exactitude de 9% à 17%. Les réseaux de neurones surpassent les SVM, et le sous-échantillonnage des mots connus améliore la prédiction.
2018	Cheriyan et al. [8]	GLM, DT, GBT	L'algorithme GBT offre la meilleure performance avec une précision de 98%, surpassant les modèles DT (71%) et GLM (64%).
2019	Yu-Sen Shih, Min-Huei Lin [21]	LSTM et analyse de sentiments	Les modèles combinés accroissent la justesse prédictive pour les ventes à court terme en utilisant l'analyse de sentiments des commentaires clients.
2021	Ulrich et al. [23]	Divers modèles de prévision (régression linéaire, GAMLSS, régression quantile, forêts de régression quantile, ARIMAX)	La performance des modèles varie selon les produits et les périodes de demande ; aucun modèle ne surpasse systématiquement les autres. La méthode de segmentation des motifs de demande améliore la précision des prévisions.
2022	Rémy Garnier [12]	ARBoost, modèle de compétitivité	Précision accrue dans les prévisions de ventes avec le modèle ARBoost, réduisant le RMSE de 15% par rapport aux modèles classiques et améliorant le MAE de 10%. Le modèle de compétitivité améliore les prévisions des parts de marché des produits concurrents.
2023	Agomoh et Ukabuiro [1]	Régression linéaire multiple	RMSE de 2,36 et MAE de 1,76, démontrant l'efficacité du modèle pour prédire les ventes avec précision.
2023	Ahmadov et Helo [2]	Réseaux neuronaux profonds	Les réseaux neuronaux profonds affichent jusqu'à 35% de gain prédictif par rapport aux méthodes classiques, avec une marge d'erreur inférieure à 7%.

TABLE 2.2 – Tableau récapitulatif des travaux connexes

Synthèse

Les travaux examinés montrent une diversité d'approches dans la prédiction des phénomènes, allant des méthodes classiques aux techniques avancées de l'apprentissage automatique. Bien que chaque méthode ait ses propres avantages et limitations, il est important de choisir celle qui convient le mieux au contexte spécifique de l'étude, en tenant compte de la disponibilité des données, de la complexité du problème et des objectifs de prédiction.

Les résultats démontrent l'importance de sélectionner et d'adapter les modèles de prévision en fonction des spécificités des données et des contextes commerciaux. Les techniques de

l'apprentissage automatique, en particulier les modèles de boosting et les réseaux neuronaux profonds, montrent des performances supérieures en termes de précision et de robustesse. L'intégration de données textuelles et l'utilisation de méta-modèles pour la sélection des techniques de prévision sont des avancées notables qui peuvent grandement améliorer la justesse des prévisions. Cependant, la complexité et le temps de calcul restent des défis à surmonter, nécessitant des compromis entre précision et efficacité opérationnelle.

Les recherches futures pourraient donc se concentrer sur des approches hybrides combinant différentes techniques pour maximiser la précision des prévisions, tout en exploitant les avantages complémentaires des méthodes classiques et modernes de l'apprentissage automatique.

Conclusion

En somme, l'apprentissage automatique constitue une avancée signification dans le domaine de l'analyse des données et de la prévision. Ce chapitre visait à offrir une vue d'ensemble complète des multiples aspects de l'apprentissage automatique en explorant les différents types d'apprentissage, les algorithmes essentiels, y compris l'apprentissage profond, ainsi que les méthodes d'optimisation et de validation des modèles. Grâce à ces connaissances, nous sommes en mesure d'utiliser des algorithmes sophistiqués pour extraire des modèles complexes à partir des données, améliorer la précision des prévisions et proposer des solutions novatrices aux défis actuels.

En transition vers le chapitre suivant, nous appliquerons ces concepts théoriques à un cas pratique de prévision des ventes chez Cevital SPA. Cela permettra de démontrer l'impact et l'efficacité des méthodes étudiées dans un contexte réel, illustrant ainsi leur potentiel pour répondre aux besoins spécifiques d'une entreprise.

3
Méthodologie de l'étude

Sommaire

Introduction **49**
3.1 Schéma de la solution proposée **49**
3.2 Environnement de développement **51**
3.3 Cas d'étude **53**
3.4 Modèle classique SARIMA **60**
3.5 Modèles d'apprentissage automatique **62**
Conclusion **66**

Introduction

Dans la suite de cette étude, nous présenterons les différents outils et techniques utilisés pour l'analyse des ventes et la mise en œuvre des modèles prédictifs. Nous expliquerons les étapes suivies pour traiter et modéliser les données, en détaillant les logiciels employés. Ensuite, nous discuterons du cadre expérimental mis en place, ainsi que de la création et de l'entraînement des modèles SARIMA, forêts aléatoires, CatBoost, XGBoost et un modèle hybride SARIMA-CatBoost.

3.1 Schéma de la solution proposée

Ce schéma offre une vue d'ensemble claire et structurée du processus de prévision des ventes, allant de l'identification des besoins à la génération des prévisions et à l'analyse des résultats. L'utilisation de Power BI pour la visualisation permet une interprétation plus facile et plus efficace des données et des résultats de la prévision. La figure 3.1 est le schéma de solution proposée.

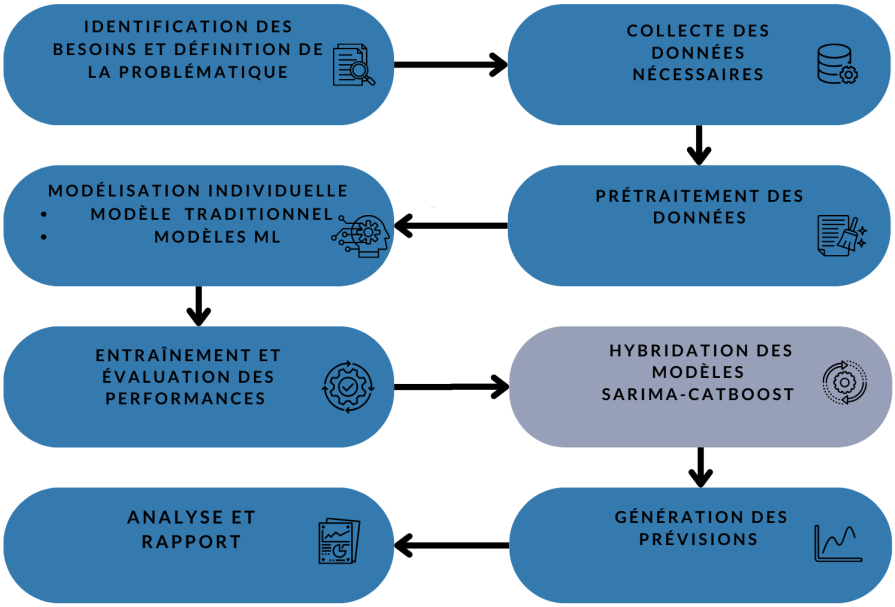

FIGURE 3.1 – Schéma de la solution proposée

- **Identification des besoins et définition de la problématique :** Cette étape initiale consiste à comprendre les besoins spécifiques de l'entreprise et à définir clairement la problématique à résoudre. Il s'agit de déterminer les objectifs de la prévision des ventes et les défis à surmonter.
- **Collecte les données nécessaires :** Après avoir défini la problématique, il est essentiel de collecter toutes les informations pertinentes afin de mener des analyses précises. Cela englobe des données de vente passées, des renseignements sur les tendances du marché.
- **Prétraitement des données :** Avant d'entamer la modélisation, il est nécessaire de procéder au nettoyage et à la préparation des données collectées. Le processus de prétraitement des données comprend de différentes étapes telles que la gestion des valeurs manquantes et la transformation des variables si nécessaire.
- **Modélisation individuelle (modèle classique et modèles ML) :** Au cours de cette étape, divers modèles de prédiction des ventes sont élaborés. Cela englobe à la fois des techniques classiques (SARIMA) et des modèles d'apprentissage automatique (tels que CatBoost). Chaque modèle est entraîné séparément pour évaluer sa performance individuelle.
- **Entraînement et évaluation des performances :** Les modèles développés ont été entraînés sur les données d'entraînement (train set) puis testés sur les données de test (test set). Des métriques appropriées ont été utilisées pour évaluer la précision des prévisions. Cette étape permet de déterminer les modèles offrant les meilleurs résultats.
- **Hybridation des modèles SARIMA et CatBoost :** Les modèles SARIMA et CatBoost ont été entraînés séparément sur les données d'entraînement. Ensuite, un méta-modèle a été utilisé pour s'entraîner sur les prévisions faites par les deux modèles. Le méta-modèle a ensuite été testé sur les données de test pour évaluer ses performances.
- **Génération des prévisions :** Les prévisions pour l'année 2024 ont été réalisées en utilisant tous les algorithmes individuellement ainsi que l'approche hybride.

- **Analyse et rapport :** Nous avons utilisé Power BI pour visualiser les données collectées sous forme de diagrammes, de graphiques et d'indicateurs clés de performance (KPI). Les prévisions générées ont également été visualisées avec Power BI, ce qui a permis une analyse détaillée des résultats.

3.2 Environnement de développement

Cette section décrit l'environnement de développement utilisé pour la mise en œuvre des modèles de prévision des ventes. Nous aborderons les outils, les bibliothèques et les technologies employés pour développer, tester et valider les modèles.

3.2.1 Langage de programmation

Pour élaborer notre pratique, nous avons opté pour l'utilisation de PYTHON.

Python est un langage de programmation open source interprété, de haut niveau et à usage général remarquable pour sa lisibilité et sa polyvalence. Créé par Guido van Rossum et publié pour la première fois en 1991. Il permet aux programmeurs d'exprimer des concepts dans moins de lignes de code que possible dans des languages comme C++ ou Java. Il prend en charge plusieurs paradigmes de programmation dont l'orienté objet, l'impératif, le fonctionnel et le procédural.

En termes de science de données et d'apprentissage automatique, Python est particulièrement apprécié pour sa syntaxe claire, sa communauté active qui contribue à une large gamme de bibliothèques et de frameworks. qui le rendent particulièrement utile pour les tâches de manipulation de données, de modélisation statistique, d'apprentissage automatique et d'apprentissage profond [17]. La figure 3.2 représente le logo Python.

FIGURE 3.2 – Logo de Python

3.2.2 Bibliothèques et frameworks pour la data science avec Python

Les outils suivants sont les fondements essentiels de la science des données avec Python, offrant des compétences indispensables pour manipuler efficacement les données, réaliser des visualisations informatives et développer des modèles d'apprentissage automatique et profond [17].

Les piliers fondamentaux

- **NumPy** : C'est une bibliothèque qui offre des tableaux multi-dimensionnels puissants et des fonctions mathématiques pour la manipulation efficace de données numériques. Il est indispensable pour le traitement de grands ensembles de données.

• **Pandas** : C'est une bibliothèque qui fournit des structures de données optimisées pour la science des données, comme les DataFrames, et des outils puissants pour l'analyse et la manipulation de données. Il facilite le nettoyage, le tri, l'agrégation et la visualisation des données.

Visualisation informative de données

• **Matplotlib** : Bibliothèque standard pour la création de graphiques et de visualisations en Python. Elle offre une large gamme de types de graphiques et d'options de personnalisation.

• **Seaborn** : C'est une bibliothèque, extension de Matplotlib, extension de Matplotlib qui construit sur ses fonctionnalités pour créer des visualisations statistiques attrayantes et informatives. Il est particulièrement utile pour la visualisation de données complexes.

• **Plotly** : Bibliothèque de visualisation de données interactive pour la création de graphiques et de visualisations riches et personnalisables. Elle offre une large gamme de types de graphiques et d'options de personnalisation, y compris des graphiques interactifs et des visualisations 3D.

Apprentissage automatique et apprentissage profond

• **Scikit-learn** : Ensemble complet d'algorithmes d'apprentissage automatique pour des tâches telles que la classification, la régression et le clustering. C'est une bibliothèque offrant une interface simple et facile à utiliser pour la mise en œuvre de modèles de ML.

• **TensorFlow/PyTorch** : Frameworks de calcul numérique de pointe pour la construction et le déploiement de modèles d'apprentissage profond. Ils offrent des fonctionnalités puissantes pour l'apprentissage automatique neuronal et les applications de deep learning.

Bibliothèques statistiques

• **Statsmodels** : Bibliothèque qui offre une large gamme de modèles statistiques et d'outils d'analyse statistique, y compris des modèles linéaires, des modèles à séries temporelles et des modèles économétriques.

• **SciPy** : C'est une bibliothèque de visualisation qui fournit des fonctions statistiques avancées pour la distribution de probabilité, les tests d'hypothèse et l'analyse statistique.

3.2.3 Environnement de développement intégré IDE

JupyterLab est un environnement de développement interactif pour créer, éditer et exécuter des documents Jupyter tels que les notebooks et les fichiers texte. Il est essentiel pour les scientifiques des données, les chercheurs et les développeurs grâce à ses extensions personnalisables et son intégration avec la ligne de commande. Il offre des fonctionnalités avancées comme la visualisation, la coloration syntaxique, et la complétion automatique pour plusieurs langages, dont Python, R et Julia [17]. La figure 3.3 représente le logo de Jupyterlab.

FIGURE 3.3 – Logo de Jupyterlab

3.2.4 Outil de visualisation et de reporting

Microsoft a développé Power BI, une série d'outils de Business Intelligence (BI) qui permet de convertir des données brutes en informations visuelles et interactives. Ces outils simplifient l'analyse des informations et la prise de décisions appropriées [18].

3.3 Cas d'étude

Dans cette section, nous présentons le cas d'étude sur la prévision des ventes au sein de l'entreprise Cevital. Nous commencerons par détailler le processus de collecte et de chargement des données, en expliquant comment les informations ont été rassemblées et organisées pour l'analyse. Ensuite, nous décrirons les différentes méthodes de modélisation utilisées pour prédire les ventes, en soulignant les algorithmes de machine learning sélectionnés et les raisons de leur choix.

3.3.1 Collecte et chargement des données

Les données de ventes en unités SKU pour les sous-familles d'un produit ont été collectées à partir des fichiers Excel fournis par le département Reporting et analyse des ventes de Cevital. Elles couvrent une période de cinq ans, allant de janvier 2019 à décembre 2023, et sont réparties par canal de distribution dans différentes régions. Un canal de distribution est un réseau structuré d'intermédiaires ou de partenaires qui facilite le déplacement des biens et services depuis le producteur jusqu'au consommateur final.

Nous avons utilisé la fonction `pd.read_excel()` de la bibliothèque `pandas` pour charger les données à partir du fichier Excel correspondant. La figure 3.4 représente l'étape du chargement.

```
df = pd.read_excel("P_BB.xlsx")
df
```

	date	Canal	Region	Sous_famille	ventes
0	2019-01-01	C1	A	P1	80.725952
1	2019-01-01	C1	A	P2	46.322000
2	2019-01-01	C1	A	P3	41.165952
3	2019-01-01	C1	B	P1	36.760256
4	2019-01-01	C1	B	P2	37.618800
...	...	...	...	...	...
1615	2023-12-01	C3	B	P2	167.108800
1616	2023-12-01	C3	B	P3	25.097600
1617	2023-12-01	C3	C	P1	11.124668
1618	2023-12-01	C3	C	P2	222.511200
1619	2023-12-01	C3	C	P3	13.664208

1620 rows × 5 columns

FIGURE 3.4 – Chargement des données

L'ensemble de données contient 1620 lignes et 5 colonnes comprenant des informations sur les ventes de trois sous-familles d'un produit notées P1, P2, P3 à travers différent canaux C1, C2, C3 dans différentes régions A, B, C.

3.3.2 Nettoyage des données

Afin d'assurer la qualité des données, il est essentiel de disposer d'une base de données de qualité, et pour cela, une vérification des valeurs manquantes a été effectuée. La figure 3.5 illustre cette opération.

```
# Vérification des valeurs manquantes dans le dataset
missing_values = df.isnull().sum()

# Affichage des valeurs manquantes
missing_values
```

```
date            0
Canal           0
Region          0
Sous_famille    0
ventes          0
dtype: int64
```

FIGURE 3.5 – Nettoyage des données

3.3.3 Prétraitement des données

Nous abordons le prétraitement des données, crucial pour préparer efficacement les informations en incluant la transformation des dates et l'extraction des caractéristiques temporelles

pertinentes.

- Transformation des dates : La colonne `date` a été convertie en type `datetime` pour faciliter l'extraction des caractéristiques temporelles.
- Extraction des caractéristiques temporelles : Des caractéristiques supplémentaires, telles que le mois et l'année, ont été extraites de la colonne `date` pour être utilisées comme variables explicatives dans le modèle.

3.3.4 Analyse exploratoire des données

L'analyse des données exploratoires (AED) constitue une étape cruciale dans la compréhension de la nature des données et dans l'identification des tendances, des modèles et des relations qui pourraient être utiles dans la modélisation des prévisions de ventes.

Cette section présente les principales analyses exploratoires réalisées sur les données de ventes d'un produit.

Statistiques descriptives

Nous utiliserons la méthode `describe()` de `pandas` comme illustrée dans la figure 3.6 pour présenter un ensemble complet de statistiques récapitulatives, incluant les moyennes, les médianes, les écarts-types et les quantiles.

```
stats_desc = df.describe()
# Affichage des statistiques descriptives du dataset
print(stats_desc)
```

```
                      date       ventes
count                 1620  1620.000000
mean   2021-06-16 06:24:00   117.277931
min    2019-01-01 00:00:00     0.516672
25%    2020-03-24 06:00:00    33.364725
50%    2021-06-16 00:00:00    83.062031
75%    2022-09-08 12:00:00   161.101062
max    2023-12-01 00:00:00  1391.468691
std                    NaN   125.652791
```

FIGURE 3.6 – Statistiques descriptives

Visualisation des données

Nous avons représenté les ventes par sous-famille, canal et région comme le montrent les figures 3.7, 3.8 et 3.9 respectivement.

Ventes totales par sous-famille

En analysant les parts de marché, il est évident que la sous-famille P2 prédomine, avec une part de marché de 40,9% pour la sous-famille P3, qui représente 34,7% des ventes, est suivie de près par la sous-famille P1, qui représente 24,4% des ventes totales.

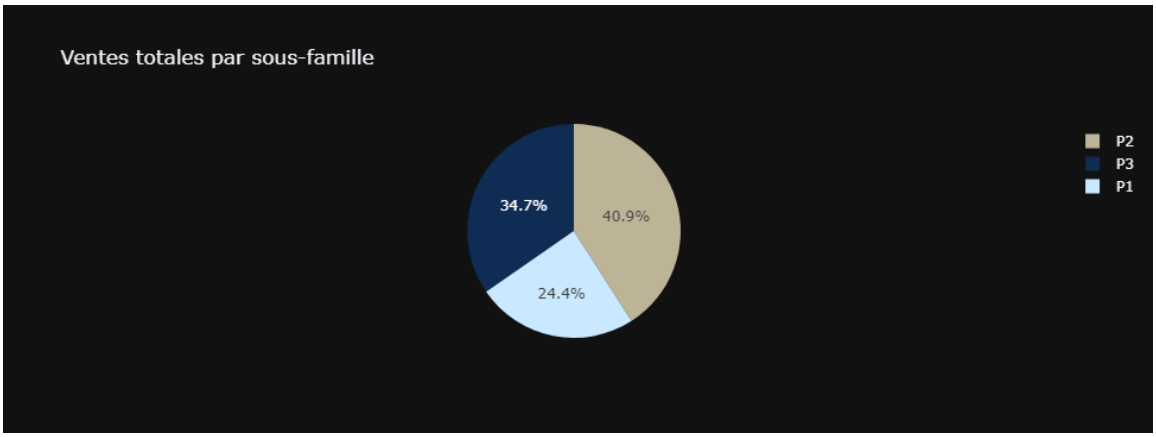

FIGURE 3.7 – Ventes totales par sous-famille

Ventes totales par canal

Selon l'analyse des ventes par canal de distribution, il ressort que le canal C2 est largement prédominant, représentant 59,7% des ventes totales. Le canal C1 a une part significative mais inférieure de 26,2% des ventes, tandis que le canal C3 représente 14,2% des ventes, ce qui en fait le canal le moins performant.

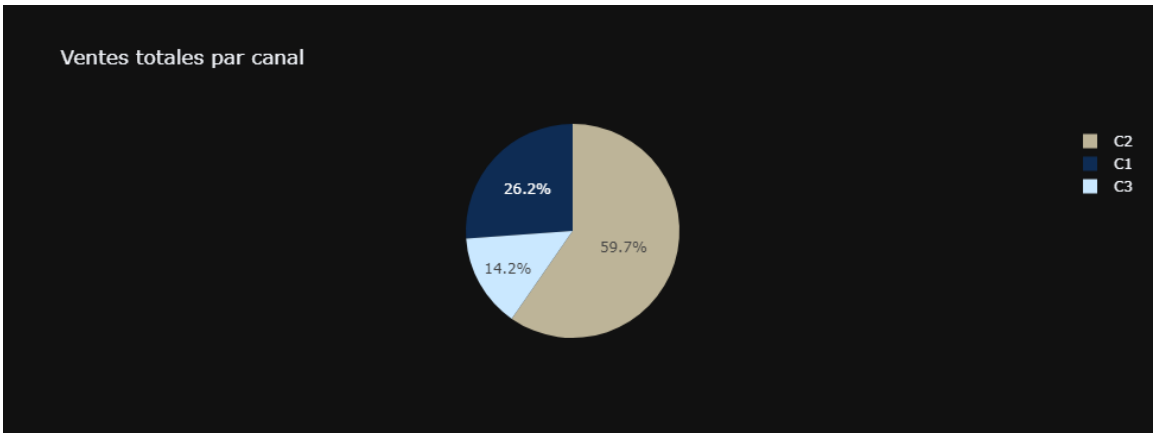

FIGURE 3.8 – Ventes totales par canal

Ventes totales par région

L'analyse des ventes par région montre que la région A est en tête avec 40,4% des ventes totales. La région B contribue de manière notable avec 37,4%, tandis que la région C ne représente que 22,2% des ventes.

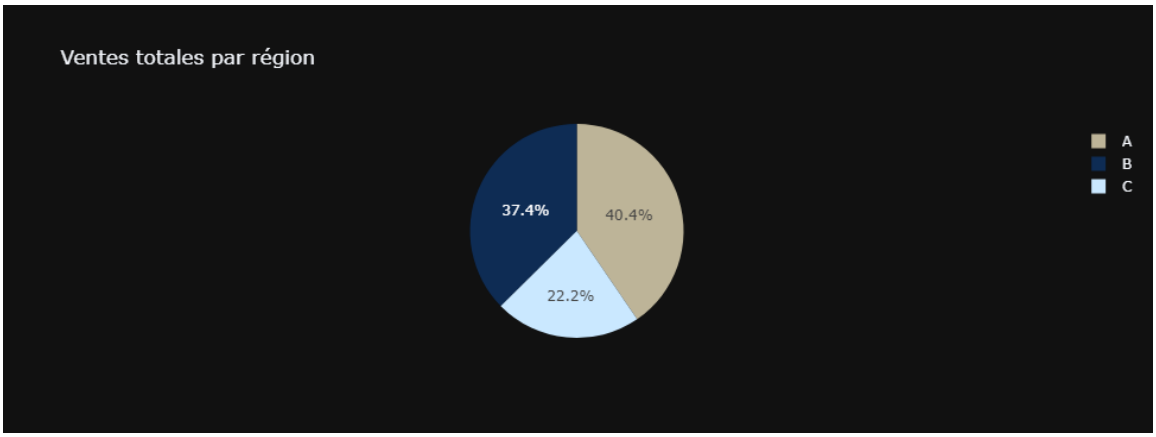

FIGURE 3.9 – Ventes totales par région

Comparaison des ventes totales

Nous analysons ici la répartition des ventes par sous-famille selon les différents canaux et régions, montrant les variations significatives dans les préférences de produits entre les canaux et les fluctuations du marché.

- **Par canal** Dans le canal C1, les ventes sont réparties de manière équilibrée entre P1, P2 et P3. Dans le canal C2, les parts de marché de P2 et P3 sont comparables, tandis que P1 est légèrement en dessous. P2 est largement prédominant dans le canal C3 indiquant une préférence marquée pour ce produit dans ce canal. La figure 3.10 fait référence à la comparaison des ventes totales par sous-famille et par canal.

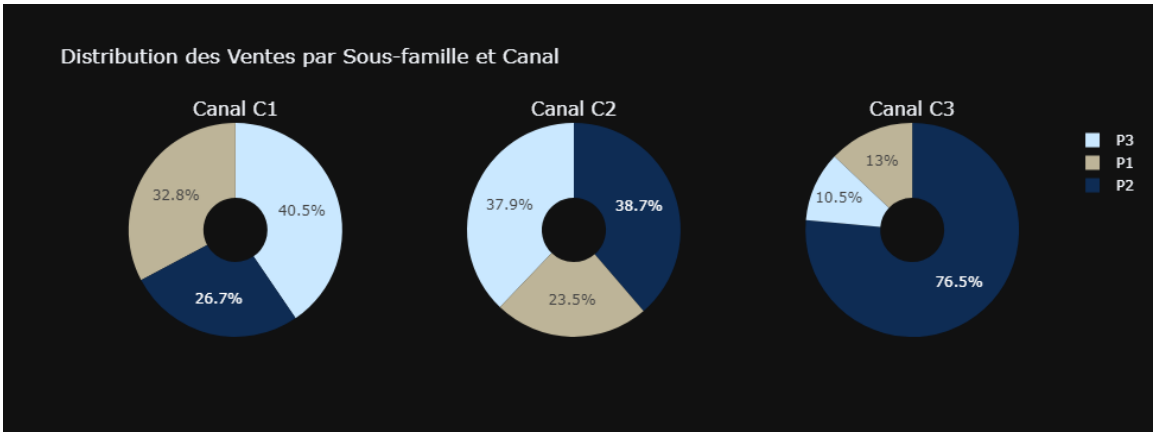

FIGURE 3.10 – Répartition des ventes par sous-famille et canal

- **Par région** Le diagramme circulaire illustre la répartition des ventes selon les sous-familles et les régions. Dans la région A, P1, P3, P2 détiennent 40%, 30,7% et 29,3% des ventes respectivement. La région B est dominée par P3 avec 42,6%, suivi de P2 avec 41,9% et P1 avec 15,6%. En ce qui concerne la région C, P2 domine avec 60,4%, tandis que P3 et P1 représentent respectivement 28,5% et 11.1%. La figure 3.11 fait référence à la comparaison des ventes totales par sous-famille et par région.

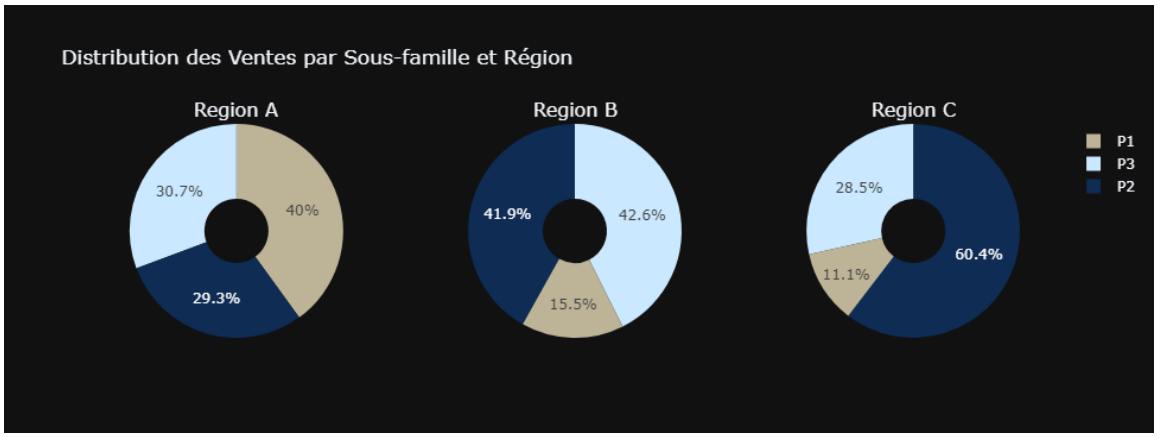

FIGURE 3.11 – Répartition des ventes par sous-famille et région

Évolution des ventes au fil du temps :

Notre objectif est d'observer les tendances et les fluctuations des ventes des sous-familles au fil du temps afin de comprendre les cycles saisonniers, les effets d'événements spécifiques, et les tendances générales, comme le montre la figure 3.12.

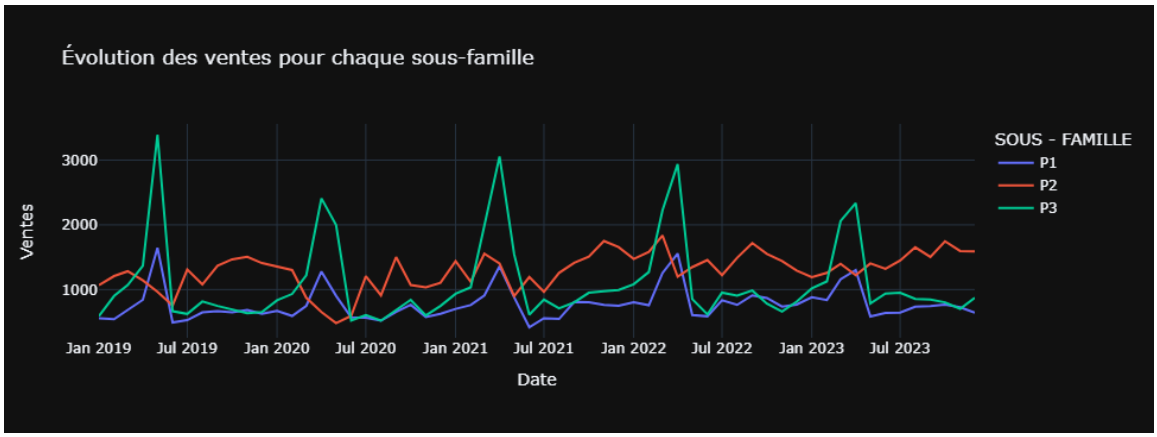

FIGURE 3.12 – Évolution des ventes

Chaque année en juillet, la sous-famille P3 connaît des pics réguliers et marqués, ce qui suggère une forte saisonnalité associée à des événements particuliers.La prévision de la sous-famille P2 est difficile en raison de la tendance à la hausse et des fluctuations saisonnières modérées. D'autre part, P1 maintient sa stabilité avec des ventes relativement faibles et peu de fluctuations, ce qui témoigne d'une demande stable mais modérée.

Avant d'entamer la modélisation des algorithmes, nous avons sélectionné une seule série chronologique (C1,B,P3) parmi toutes les séries disponibles afin de simplifier notre approche.

3.3.5 Décomposition saisonnière

Pour comprendre les tendances et les saisons dans les données de ventes, une décomposition des séries temporelles a été réalisée . La figure 3.13 illustre cette décomposition.

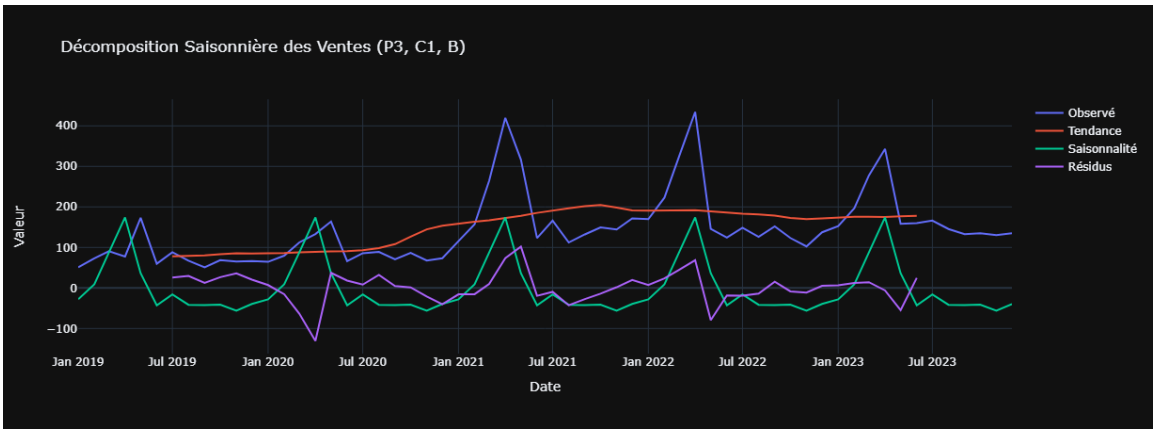

FIGURE 3.13 – Décomposition saisonnière

La décomposition saisonnière présente les ventes de la sous-famille P3 dans le canal C1 et la région B. Les différentes composantes de la série temporelle sont décomposés pour analyser plus finement les variations des ventes. Voici une interprétation de chaque composante du graphique :

- **Observé (bleu) :** Montre les ventes réelles avec des fluctuations importantes et des pics réguliers, indiquant une forte saisonnalité.
- **Tendance (rouge) :** Indique une croissance des ventes jusqu'à mi-2022, suivie d'une stabilisation. Cela reflète une phase de croissance initiale puis une maturité du marché.
- **Saisonnalité (vert) :** Illustre des variations périodiques avec des pics réguliers en début d'année, correspondant probablement à des périodes de forte demande saisonnière.
- **Résidus (violet) :** Montre les variations aléatoires non expliquées par la tendance ou la saisonnalité. Les pics et creux peuvent signaler des événements spécifiques ou des anomalies tels que des promotions exceptionnelles ou l'impact des facteurs externes.

3.3.6 Traitement des données

Avant de procéder à la modélisation, certaines étapes de prétraitement des données sont nécessaires pour garantir la qualité des données et la précision des modèles.

Transformation logarithmique des données

Pour stabiliser la variance des données de ventes, une transformation logarithmique utilisant la fonction `log1p` a été appliquée afin d'améliorer les performances des modèles d'apprentissage automatique. Cela permet de réduire l'impact des valeurs extrêmes et de rendre les données plus adaptées à la modélisation. Après l'entraînement des modèles, les valeurs prédites sont remises à leur échelle originale en appliquant la transformation inverse `np.expm1()`.

Division des données

Pour évaluer correctement les modèles, les données transformées ont ensuite été divisés en périodes d'entraînement 80% et de test 20% en respectant l'ordre chronologique. Les périodes d'entraînement et de test ont été définies comme suit :

- Période d'entraînement : Du 1er janvier 2019 au 1er décembre 2022.
- Période de test : Du 1er janvier 2023 au 1er décembre 2023.

Les données utilisées pour l'apprentissage ainsi que pour le test sont les mêmes pour l'ensemble des algorithmes, le choix de la division 80/20 se justifie par les raisons suivantes :

- La division 80/20 est une convention largement adoptée dans l'apprentissage machine, facilitant la comparaison des performances entre modèles.
- Cette pratique réduit le risque de surapprentissage en fournissant suffisamment de données pour l'entraînement tout en réservant une portion significative pour l'évaluation indépendante du modèle.
- Adaptée à notre petit dataset, cette division a été choisie après avoir exploré différentes options, offrant un équilibre optimal entre l'entraînement du modèle et l'évaluation de sa capacité de généralisation.

3.4 Modèle classique SARIMA

Cette phase se concentre sur la mise en œuvre des modèles de prévision des ventes en utilisant différentes modèles de prévision.

Le choix du modèle SARIMA a été motivé par diverses raisons. En premier lieu, ce modèle convient parfaitement pour représenter les séries temporelles qui présentent des tendances et des saisonnalités, caractéristiques des données de ventes qui présentent des fluctuations régulières tout au long de l'année. De plus, il offre une grande souplesse et une puissance d'analyse pour des données complexes, offrant ainsi une grande une grande flexibilité et une puissance d'analyse. Les différentes étapes de la modélisation avec SARIMA incluent l'analyse de la stationnarité, la sélection des paramètres optimaux et l'ajustement du modèle afin d'obtenir les prévisions les plus précises.

Test de stationnarité

Pour vérifier si la série temporelle est stationnaire, nous avons utilisé le test de Dickey-Fuller augmenté (ADF).

Analyse ACF et PACF

Pour identifier les ordres des composants AR et MA graphiquement, nous avons utilisé les fonctions d'autocorrélation (ACF) et d'autocorrélation partielle (PACF) comme l'illustre la figure 3.14.

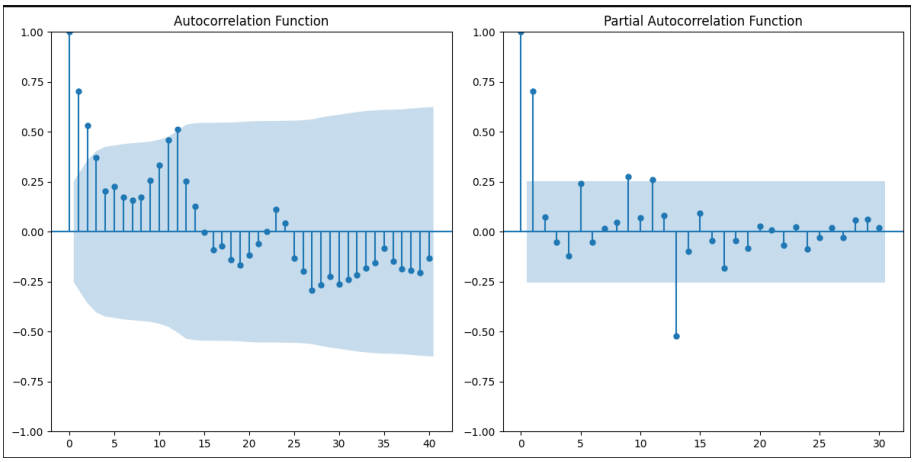

FIGURE 3.14 – Fonctions ACF et PACF

Ajustement des hyperparamètres

Afin d'automatiser le processus d'ajustement des paramètres du modèle SARIMA, la fonction `auto_arima` a été employée . Celle-ci différencie automatiquement la série si nécessaire pour la rendre stationnaire, et minimise le critère d'information d'Akaike (AIC). Le faible AIC indique que ce modèle offre un bon équilibre entre complexité et précision.

Les hyperparamètres $(0, 1, 1)$ pour le modèle ARIMA et $(1, 0, 0, 12)$ pour la composante saisonnière ont été sélectionnés.

- $p = 0$: L'absence de termes auto-régressifs indique que les valeurs passées ne sont pas directement utilisées pour la prédiction, simplifiant ainsi le modèle et réduisant le risque de surajustement.
- $d = 1$: Une différenciation de premier ordre est suffisante pour rendre la série temporelle stationnaire.
- $q = 1$: Utiliser un terme de moyenne mobile permet de capturer les erreurs de prédiction précédentes, améliorant ainsi la précision du modèle.
- $P = 1$: Un terme auto-régressif saisonnier indique que la valeur actuelle est influencée par la valeur de la même période de l'année précédente, capturant ainsi les tendances saisonnières.
- $D = 0$: Aucune différenciation saisonnière supplémentaire n'est nécessaire, ce qui simplifie le modèle.
- $Q = 0$: L'absence de termes de moyenne mobile saisonniers réduit la complexité du modèle.
- $S = 12$: La période saisonnière de 12 mois est appropriée pour les données mensuelles, capturant les effets saisonniers annuels.

Entraînement du modèle

Le modèle SARIMA a été créé et entraîné en utilisant les meilleurs paramètres identifiés par la fonction `mod.fit`.

3.5 Modèles d'apprentissage automatique

Après avoir bien analysé notre dataset et comparé entre divers algorithmes de régression, nous avons désigné les plus adaptés à notre problématique. On cite :

- Forêt aléatoire
- XGBoost
- CatBoost
- Hybridation SARIMA-CatBoost

3.5.1 Forêt aléatoire

Le modèle forêt aléatoire a été choisi en raison de sa capacité à améliorer la précision en combinant plusieurs arbres de décision, de sa résistance au surapprentissage, de sa gestion efficace des relations non linéaires et des valeurs manquantes, ainsi que de sa stabilité et de sa performance même avec des jeux de données bruyants.

Grille des hyperparamètres

Nous avons défini la grille des hyperparamètres suivante pour le modèle forêt aléatoire :

- `n_estimators :` Les valeurs testées sont $[100, 200, 300]$.
- `max_depth :` Les valeurs testées sont $[3, 5, 7, 10]$.
- `min_samples_split :` Les valeurs testées sont $[2, 5, 10]$.
- `min_samples_leaf :` Les valeurs testées sont $[1, 2, 4]$.
- `max_features :` Les valeurs testées sont ['auto', 'sqrt', 'log2'].

Meilleurs paramètres sélectionnés

Après avoir exécuté `GridSearchCV` sur les données d'entraînement, les meilleurs hyperparamètres sélectionnés étaient les suivants :

- `n_estimators = 100 :` Le modèle utilise 100 arbres pour faire des prédictions. Ce nombre est un bon compromis entre performance et temps de calcul.
- `max_depth = 10` : La profondeur maximale des arbres est fixée à 10. Cette profondeur permet de capturer des relations complexes dans les données sans trop de surajustement.
- `min_samples_split = 2 :` Le nombre minimum d'échantillons requis pour diviser un nœud est de 2. Cela permet à l'arbre de se développer complètement sans restriction excessive.
- `min_samples_leaf = 1 :` Le nombre minimum d'échantillons requis pour être à une feuille est de 1, ce qui permet de capturer des variations locales fines.
- `max_features = 'sqrt' :` Le nombre maximum de caractéristiques à considérer pour chaque division est la racine carrée du nombre total de caractéristiques, on réduit la variance sans augmenter significativement le biais, ce qui est souvent un bon choix par défaut pour les forêts aléatoires.

Création et entraînement du modèle

Le modèle forêt aléatoire pour la régression a été initialisé via `RandomForestRegressor()`, puis entraîné sur les données d'apprentissage grâce à la fonction `fit()` de l'objet modèle `best_rf_model`.

3.5.2 XGBoost

Il a été sélectionné en raison de ses performances remarquables en matière de précision et de vitesse, de sa capacité à gérer des interactions complexes entre les variables, de sa résistance au surapprentissage grâce à la régularisation intégrée, et de ses options avancées de contrôle des hyperparamètres qui permettent une optimisation précise du modèle.

Grille des hyperparamètres

Nous avons configuré la grille d'hyperparamètres suivante pour le modèle XGBoost :

- `n_estimators` : Les valeurs testées sont [50, 100, 200, 300].
- `learning_rate` : Les valeurs testées sont [0.01, 0.1, 0.2].
- `max_depth` : Profondeur maximale de l'arbre. Les valeurs testées sont [3, 5, 7].
- `gamma` : Les valeurs testées sont [0, 0.1, 0.2].
- `subsample` : Les valeurs testées sont [0.7, 0.8, 1.0].
- `colsample_bytree` : Les valeurs testées sont [0.7, 0.8, 0.9, 1.0].

Meilleurs paramètres sélectionnés

Après avoir exécuté `GridSearchCV` sur les données d'entraînement, les hyperparamètres optimaux sélectionnés étaient les suivants :

- `n_estimators = 50` : Le modèle utilise 50 arbres pour faire des prédictions. Ce nombre est un bon équilibre entre performance et temps de calcul.
- `learning_rate = 0.1` : Un taux d'apprentissage de 0.1 permet au modèle de converger rapidement sans risquer de surajuster les données.
- `max_depth = 5` : La profondeur maximale des arbres est fixée à 5. Cette profondeur permet de capturer des relations complexes dans les données tout en évitant le surajustement.
- `gamma = 0` : Aucune réduction de perte minimale requise pour effectuer une partition. Cela permet à l'algorithme de considérer toutes les partitions possibles.
- `subsample = 0.7` : 70% des échantillons sont utilisés par le modèle pour entraîner chaque arbre, ce qui permet d'éviter le surajustement.
- `colsample_bytree = 1.0` : Le modèle utilise toutes les caractéristiques disponibles pour chaque arbre, maximisant ainsi les informations utilisées pour les divisions.

Création et entraînement du modèle

La fonction `XGBRegressor()` a été utilisée pour créer un modèle XGBoost de régression, qui a ensuite été entraîné avec la fonction `best_xgb_model.fit()` sur les données d'entraînement.

3.5.3 CatBoost

Son choix a été fait en raison de ses performances exceptionnelles, de sa capacité à gérer automatiquement les variables catégorielles sans prétraitement, de son efficacité en calcul, de sa résistance au surapprentissage grâce à la régularisation, et de sa capacité à gérer des relations non linéaires complexes tout en exploitant les interactions entre les caractéristiques pour améliorer les prédictions.

Grille des hyperparamètres

Nous avons établi la grille d'hyperparamètres suivante pour le modèle CatBoost :

- `iterations` : Les valeurs testées sont [100, 200, 300].
- `learning_rate` : Les valeurs testées sont [0.01, 0.1, 0.2].
- `depth` : Les valeurs testées sont [3, 5, 7].
- `l2_leaf_reg` : Les valeurs testées sont [1, 3, 5].
- `border_count` : Les valeurs testées sont [32, 50, 100].

Meilleurs paramètres sélectionnés

Après avoir exécuté `GridSearchCV` sur les données d'entraînement, les meilleurs hyperparamètres sélectionnés étaient les suivants :

- `border_count = 32` : Nombre de frontières pour les caractéristiques numériques. Une valeur de 32 permet de capturer suffisamment de complexité sans surcharger le modèle.
- `depth = 3` : La profondeur maximale des arbres est fixée à 3. Cette profondeur permet de capturer des relations complexes dans les données tout en évitant le surajustement.
- `iterations = 300` : Le modèle utilise 300 itérations pour faire des prédictions. Ce nombre est un bon compromis entre performance et temps de calcul.
- `l2_leaf_reg = 1` : Coefficient de régularisation L2 pour les feuilles, ce qui aide à prévenir le surajustement en pénalisant les feuilles trop complexes.
- `learning_rate = 0.1` : Un taux d'apprentissage de 0.1 permet au modèle de converger rapidement sans risquer de surajuster les données.

Création et entraînement du modèle

Nous avons employé la fonction `CatBoostRegressor()` pour créer le modèle et utilisé la fonction `best_cat_model.fit` pour entraîner celui-ci.

3.5.4 Hybridation SARIMA-CatBoost

Pour affiner la précision des prévisions de ventes, nous avons adopté une approche de stacking en testant plusieurs combinaisons d'algorithmes, constatant que l'hybridation SARIMA-CatBoost a produit de meilleurs résultats et a tiré parti des atouts de chaque modèle. SARIMA a capturé les dépendances temporelles et saisonnières, tandis que CatBoost, enrichi de caractéristiques temporelles (année, mois, jour). Les prédictions de ces deux modèles ont été fusionnées

dans un modèle de méta-apprentissage (régression linéaire) pour générer des prévisions finales plus robustes. Cette méthodologie a permis de réduire l'erreur globale et d'améliorer la performance prédictive.

Ajustement automatique des hyperparamètres

Les paramètres sélectionnés pour le modèle SARIMA via l'algorithme d'optimisation `auto_arima` et CatBoost via `GridSearchCV` sont les mêmes que ceux définies précédemment pour la série en question.

Étapes de la modélisation hybride

- Entraînement du modèle SARIMA : Après avoir identifié les paramètres optimaux par `auto_arima`, nous avons ajusté un modèle `SARIMAX` en utilisant ces paramètres optimaux. Le modèle SARIMA a été entraîné sur les données de la série temporelle transformée logarithmiquement pour capturer les motifs saisonniers et temporels. L'entraînement du modèle se fait avec la commande `mod.fit()`.
- Prévisions avec SARIMA : Le modèle ajusté a été utilisé employé afin de prédire l'ensemble des tests avec la commande `get_prediction()`. Les prédictions obtenues en échelle logarithmique ont été converties en échelle originale (ventes) en utilisant la fonction exponentielle inverse `np.expm1()`.
- Enrichissement des données pour CatBoost : Pour CatBoost, nous avons ajouté des caractéristiques temporelles supplémentaires aux données d'entraînement et de test (année, mois, jour). Cela a permis au modèle CatBoost de mieux capturer les variations saisonnières et temporelles dans les données.
- Entraînement et prévisions avec CatBoost : En utilisant les hyperparamètres les plus performants trouvés grâce à `GridSearchCV`, nous avons entraîné un modèle CatBoost sur les données enrichies avec la fonction `best_catboost_model.fit()`. Les prévisions de CatBoost ont ensuite été obtenues sur l'ensemble de test avec la fonction `predict()`, puis également transformées de l'échelle logarithmique à l'originale avec la fonction `np.expm1()`.
- Combinaison des prédictions avec méta-apprentissage : En utilisant un modèle de méta-apprentissage basé sur la régression linéaire, les prédictions des modèles SARIMA et CatBoost ont été combinées.
    1. Préparation des données pour le méta-apprentissage : Les prévisions des modèles SARIMA et CatBoost ont été intégrées dans un ensemble de données de stacking pour l'ensemble de test. Ce nouvel ensemble de données a deux colonnes principales : une pour les prédictions de SARIMA et une autre pour les prédictions de CatBoost.
    2. Entraînement du modèle de méta-apprentissage : Nous avons utilisé les prévisions de SARIMA et de CatBoost comme caractéristiques pour entraîner un modèle de régression linéaire grâce à la commande `meta_model.fit`. Le modèle de méta-apprentissage a été ajusté pour minimiser l'erreur entre les prévisions combinées et les valeurs de ventes réelles. En d'autres termes, nous avons cherché à trouver

la meilleure combinaison linéaire des deux séries de prédictions pour obtenir une estimation finale plus précise des ventes.

3. Génération des prévisions finales : Une fois le modèle de régression linéaire entraîné, nous l'avons utilisé pour générer des prévisions finales en appliquant les coefficients de régression aux prévisions de SARIMA et de CatBoost, la commande `meta_model.predict` a été utilisée. Les prédictions finales sont donc une combinaison pondérée des prédictions des deux modèles de base.

Conclusion

En conclusion, nous avons décrit le processus de collecte, de nettoyage et de prétraitement des données, essentiels pour garantir la qualité et la fiabilité des modèles prédictifs. Précisément nous avons exploré les modèles SARIMA, forêt aléatoire, XGBoost et CatBoost afin d'évaluer leurs performances respectives. Par la suite, nous avons proposé un modèle hybride SARIMA-CatBoost. Dans le prochain chapitre, nous analyserons en détail les résultats obtenus et discuterons de leur implication pour la prévision des ventes.

4
Résultats et discussions

Sommaire

Introduction **67**
4.1 Présentation des résultats **67**
4.2 Évaluation de divers algorithmes sur plusieurs séries chronologiques . . **75**
4.3 Tableau de bord **76**
Conclusion **78**

Introduction

Dans ce dernier chapitre, nous présenterons les résultats obtenus à partir des différents algorithmes de prévision des ventes appliqués aux données de la combinaison (C1,B,P3). Nous discuterons des performances de chaque modèle en utilisant les métriques d'évaluation telles que l'erreur absolue moyenne MAE, la racine carrée de l'erreur quadratique moyenne RMSE et le coefficient de détermination R^2. Les algorithmes considérés sont SARIMA, forêt aléatoire, XGBoost, CatBoost et le modèle hybride SARIMA-CatBoost.

En outre, nous comparerons les performances de ces modèles sur différentes séries chronologiques et présenterons un tableau de bord développé grâce a Power BI pour visualiser les résultats.

4.1 Présentation des résultats

Cette section présente les résultats des modèles de prévision des ventes, mettant en avant les performances des modèles SARIMA, forêt aléatoire, XGBoost, CatBoost et SARIMA-CatBoost.

4.1.1 Prévisions avec le modèle SARIMA

Le modèle entraîné a été utilisé pour prévoir les ventes sur l'ensemble de test, en retransformant les prédictions logarithmiques pour obtenir les valeurs réelles. Les prévisions marquées en rouge ont été générées à l'aide de la fonction `get_prediction`.

Pour l'année 2024, les projections des ventes, indiquées en vert, ont été calculées mensuellement en utilisant la fonction `pd.date_range` pour créer une série chronologique. Cette série a été transformée en un ataFrame intégrant les mois et les années comme variables explicatives. En appliquant les meilleurs paramètres du modèle SARIMA, les ventes pour 2024 ont été estimées avec la fonction `get_prediction`. La figure 4.1 démontre les prévisions générées sur l'ensemble de test et sur l'année 2024.

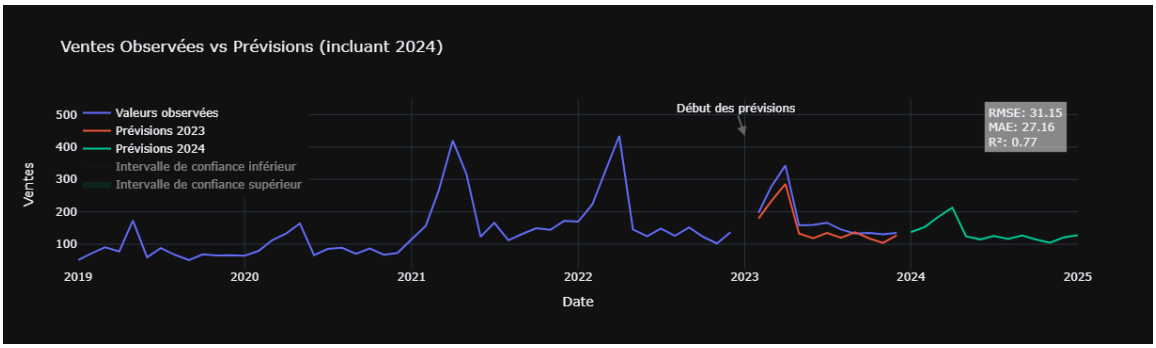

FIGURE 4.1 – Résultats avec SARIMA

Interprétation des métriques d'évaluation

- Un RMSE de 31.15 indique que l'écart quadratique moyen entre les valeurs prévues et les valeurs observées est d'environ 31.15. Le RMSE, étant plus sensible aux grandes erreurs, suggère que certaines prévisions peuvent s'écarter de manière significative des valeurs réelles. Cela montre que, même si le modèle SARIMA peut gérer les variations générales, il reste sensible à quelques fluctuations importantes.
- Un MAE de 27.16 signifie que, en moyenne, les prévisions du modèle s'écartent de 27.16 unités des valeurs observées. Cela montre un niveau d'erreur absolue moyen acceptable pour des prévisions de ventes.
- Un R^2 de 0.77 indique que 77% de la variance des ventes observées est expliquée par le modèle de prévision. Cela démontre une bonne capacité prédictive du modèle, bien que 22.58% de la variance ne soit pas expliquée par le modèle.

Interprétation des prévisions pour 2024

- Les prévisions pour 2024 semblent indiquer une stabilité des ventes avec des valeurs qui oscillent autour de la moyenne des périodes précédentes. Il n'y a pas de pics significatifs comme ceux observés en 2021 ou 2023.
- Étant donné les bonnes performances du modèle sur la période de test, on peut raisonnablement faire confiance aux prévisions pour 2024, à moins que des changements majeurs n'interviennent dans le marché.

4.1.2 Prévisions avec le modèle forêt aléatoire

Le modèle entraîné a été déployé pour prédire les ventes sur l'ensemble de test, avec inversion des prédictions logarithmiques pour obtenir les valeurs originales. Les prévisions en rouge ont été obtenues à l'aide de la fonction `best_rf_model.predict`.

Pour l'année 2024, les prévisions de ventes en vert ont été générées mensuellement en utilisant la fonction `pd.date_range` pour créer une série temporelle, transformée en un DataFrame `X_2024` avec mois et années comme caractéristiques. Le modèle optimisé a été utilisé pour prédire les ventes de 2024 avec la fonction `best_rf_model.predict(X_2024)`. La figure 4.2 illustre les prévisions sur l'ensemble de test et pour l'année 2024.

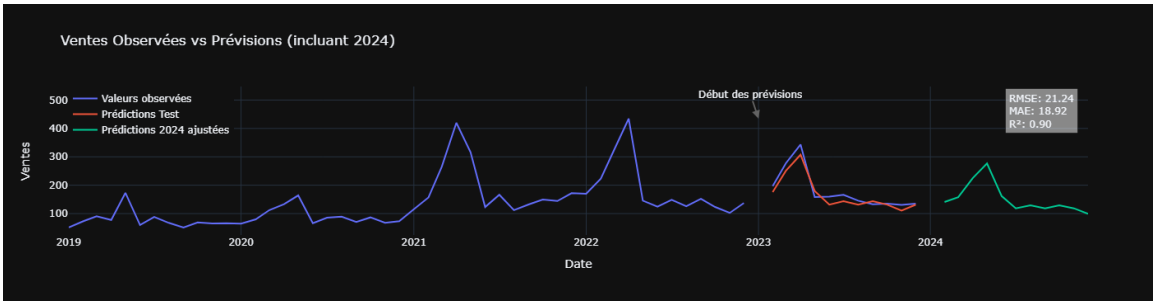

FIGURE 4.2 – Résultats avec forêt aléatoire

Interprétation des métriques d'évaluation

- Un RMSE de 21.24 indique des écarts quadratiques moyens relativement faibles entre les prévisions et les observations, suggérant une précision élevée.
- Avec un MAE de 18.92, les prévisions s'écartent en moyenne de 16.30 unités par rapport aux valeurs réelles, montrant une faible erreur absolue.
- Un R^2 de 0.9 révèle que 90% de la variance des ventes est expliquée par le modèle, indiquant une forte capacité explicative..

Interprétation des prévision 2024

Voici quelques observations concernant les valeurs prédites pour l'année 2024.

- Les valeurs prédites continuent sur une trajectoire similaire aux années précédentes, sans variations majeures.
- Les excellentes performances du modèle sur la période de test suggèrent une fiabilité élevée pour les prévisions de 2024, sauf en cas de changements significatifs du marché.

4.1.3 Prévisions avec le modèle XGBoost

L'algorithme formé a été employé pour prédire les ventes sur l'ensemble de test en convertissant les prédictions logarithmiques pour retrouver les valeurs réelles. Les prévisions marquées en rouge ont été obtenues à l'aide de la fonction `best_xgb_model.predict`.

Pour l'année 2024, les prévisions de ventes annotées en vert ont été calculées mensuellement en utilisant la fonction `pd.date_range` pour créer une série chronologique, convertie ensuite en un DataFrame `X_2024` avec les mois et les années comme variables explicatives. Le modèle ajusté a été utilisé pour estimer les ventes de 2024 à l'aide de la fonction `best_xgb_model.predict(X_2024)`. La figure 4.3 illustre les prévisions obtenues sur l'ensemble de test et sur l'année 2024.

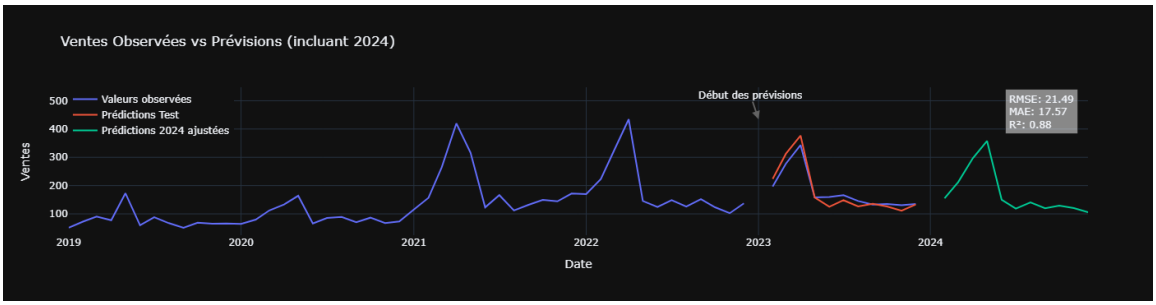

FIGURE 4.3 – Résultats avec XGBoost

Interprétation des métriques d'évaluation

- Un RMSE de 21.49 est un peu plus élevé que celui du modèle forêt aléatoire (21.24), mais reste inférieur à celui du premier modèle SARIMA. Cela indique que les prévisions de XGBoost sont précises mais avec une légère augmentation des erreurs quadratiques par rapport au modèle forêt aléatoire.
- Un MAE de 17.57, plus faible que le MAE de forêt aléatoire (18.92), montre tout de même que les prévisions s'écartent de manière raisonnable des valeurs observées.
- Un R^2 de 0.88 démontre une bonne capacité explicative, légèrement inférieure à celle du modèle forêt aléatoire (0.9), mais nettement supérieure à celle du modèle SARIMA.

Interprétation des prévision 2024

- Les prévisions pour 2024 indiquent une continuité stable des ventes avec des variations mineures autour de la moyenne historique, similaire aux autres modèles.
- Les bonnes performances sur la période de test suggèrent que les prévisions de XGBoost pour 2024 sont fiables, à l'exception de changements majeurs sur le marché.

4.1.4 Prévisions avec le modèle CatBoost

Le modèle entraîné a été utilisé pour anticiper les ventes sur l'ensemble de test en renversant les prédictions logarithmiques pour obtenir les valeurs réelles. Les prévisions marquées en rouge ont été générées à l'aide de la fonction `best_cb_model.predict`.

Pour l'année 2024, les prévisions de ventes, marquées en vert, ont été établies mensuellement en utilisant la fonction `pd.date_range` pour créer une série temporelle, transformée ensuite en un DataFrame `X_2024` comprenant les mois et les années comme variables distinctives. En utilisant les meilleurs hyperparamètres du modèle, les ventes de 2024 ont été prédites à

l'aide de la fonction `best_cb_model.predict(X_2024)`. La figure 4.4 illustre les prévisions générées sur l'ensemble de test et sur l'année 2024.

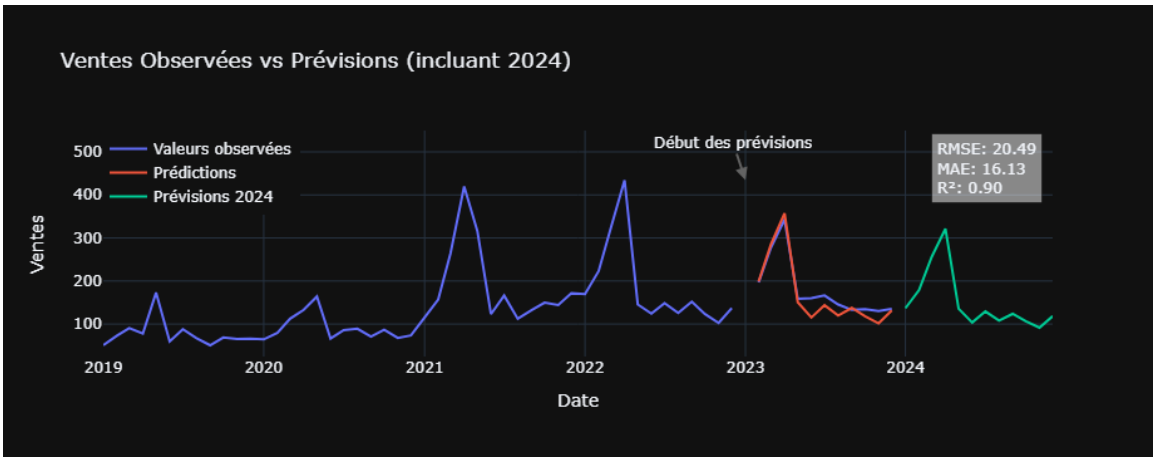

FIGURE 4.4 – Résultats avec CatBoost

Interprétation des métriques d'évaluation

- Un RMSE de 20.49 indique que, en moyenne, les prévisions s'écartent de 20.49 unités des valeurs réelles. Bien que les données de vente soient complexes, le modèle réussit à fournir des estimations cohérentes.
- La précision du modèle CatBoost est mise en évidence par le MAE de 16.13, avec des prévisions qui ne diffèrent en moyenne que de 16.13 unités des valeurs réelles. Le modèle est capable de détecter les subtiles variations des données de vente, ce qui lui permet d'offrir des informations précieuses pour la prise de décision.
- La forte valeur de 0,90 du coefficient de détermination suggère la capacité de CatBoost à expliquer 90% de la variation des ventes observée. La performance du modèle met en évidence son efficacité dans l'identification des schémas et des tendances sous-jacentes, ce qui en fait un outil puissant pour prévoir les comportements futurs du marché.

Interprétation des prévisions pour 2024

- La variance des ventes projetée pour 2024 est inférieure aux données historiques.
- La transition en douceur des prédictions 2023 aux prévisions 2024 indique une cohérence dans le modèle, suggérant qu'il n'y a pas de rupture soudaine ou de comportement inattendu.
- Le modèle semble donc avoir reproduit les tendances saisonnières présentes dans les données historiques et les a intégrées dans les prévisions pour 2024.

4.1.5 Prévisions avec le modèle hybride SARIMA-CatBoost

Nous présentons ici les résultats des prévisions sur l'ensemble de test en rouge obtenues en combinant les modèles SARIMA et CatBoost, chacun préalablement entraîné. Le modèle

SARIMA ajusté, a utilisé la fonction `get_prediction` pour générer des prévisions de ventes, qui ont ensuite été converties en valeurs réelles. Parallèlement, un modèle CatBoost optimisé a généré des prévisions à l'aide de la fonction `best_catboost_model.predict`, également converties en valeurs réelles. Les prévisions des deux modèles ont ensuite été combinées à l'aide du métamodèle régression linéaire, dont les résultats finaux ont été obtenus grâce à la fonction `meta_model.predict`.

Pour l'année 2024, les prévisions de ventes en en vert ont été effectuées en utilisant l'hybridation des deux modeles. Les dates mensuelles de 2024 ont été générées et transformées en un DataFrame. Les ventes ont ensuite été prédites séparément par les modèles CatBoost et SARIMA, puis converties en valeurs réelles. Enfin, les prévisions des deux modèles ont été fusionnées à l'aide d'un métamodèle de régression linéaire pour obtenir les prévisions finales. La figure 4.5 présente les prévisions sur l'ensemble de test et pour l'année 2024.

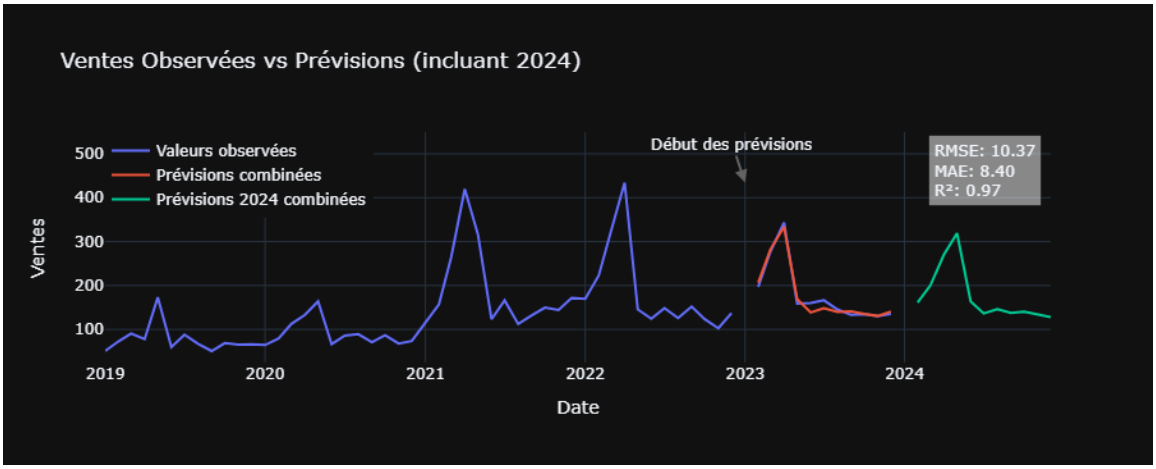

FIGURE 4.5 – Résultats avec le modèle hybride SARIMA-CatBoost

Interprétation des métriques d'évaluation

- Le RMSE de 10.37 est significativement plus bas que ceux des autres modèles, indiquant des prévisions précises avec des erreurs quadratiques moyennes réduites.
- Un MAE de 8.40 montre que les prévisions s'écartent en moyenne de 8.40 unités des valeurs observées, ce qui représente une amélioration notable par rapport aux autres modèles.
- Un R^2 de 0.97 révèle que 97.18% de la variance des ventes est expliquée par le modèle hybride, indiquant une capacité explicative élevée.

En résumé, les métriques de performance suggère la précision et la fiabilité du modèle hybride CatBoost-SARIMA pour les prévisions de ventes de l'année 2023, ce qui en fait une solution prometteuse pour améliorer la planification et la prise de décision stratégique dans le domaine commercial.

Interprétation des prévisions pour 2024

- Les prévisions pour 2024 montrent une transition cohérente, sans anomalies soudaines, suggérant une continuité et une stabilité dans le modèle.

- Les valeurs observées montrent des fluctuations avec des pics périodiques et des périodes de stabilité. Les prévisions pour 2024 suivent un schéma similaire, ce qui indique que le modèle a bien appris les tendances historiques.
- Les performances sur la période de test suggèrent que les prévisions de l'hybridation CatBoost et SARIMA pour 2024 sont fiables.

4.1.6 Évaluation comparative des modèles

Pour évaluer l'efficacité des différents modèles de prévision, nous avons comparé leurs performances en termes des métriques RMSE, MAE , R^2, et interprétation des prévisions pour l'année 2024. Le tableau 4.1 présente les résultats de cette comparaison, mettant en évidence les forces et les faiblesses de chaque modèle. Les figures 4.6 et 4.7 présentent la performance des différents algorithmes sous forme de diagrammes à barres.

Modèle	RMSE	MAE	R^2	Interprétation des prévisions pour 2024
SARIMA	31.15	27.16	0.77	Captures les tendances saisonnières, mais moins fiable.
Forêt aléatoire	21.24	18.92	0.90	Stabilité avec légères variations, robuste.
XGBoost	21.49	17.57	0.88	Bonne sensibilité, continuité stable.
CatBoost	20.49	16.13	0.90	Bonne robustesse face aux fluctuations saisonnières.
SARIMA-CatBoost	10.37	8.40	0.97	Prévisions les plus stables et précises, très robuste.

TABLE 4.1 – Tableau comparatif des performances des modèles de prévision

L'analyse des différents modèles de prévision des ventes suggèrent des performances variées en termes de précision et de robustesse. Le modèle SARIMA, bien qu'efficace pour capturer les tendances saisonnières, présente des erreurs plus élevées (RMSE de 31.15 et MAE de 27.16), le rendant moins fiable comparé aux autres. Les modèles forêt Aléatoire et XGBoost améliorent la précision avec des RMSE respectifs de 21.24 et 21.49, et des MAE de 18.92 et 17.57, offrant une stabilité et une bonne sensibilité aux variations. CatBoost se distingue par sa robustesse face aux fluctuations saisonnières, affichant une meilleure précision (RMSE de 20.49 et MAE de 16.13). Toutefois, la combinaison SARIMA-CatBoost surpasse les autres modèles, avec des valeurs de RMSE (10.37) et de MAE (8.40) inférieures, et un coefficient de détermination élevé (0.97). Cette combinaison offre des prévisions plus précises et plus stables, intégrant la capacité de SARIMA à saisir les tendances saisonnières et la puissance prédictive de CatBoost. Ces prévisions, basées uniquement sur les données disponibles et n'incluant pas de facteurs externes, ont donné des résultats convaincants.

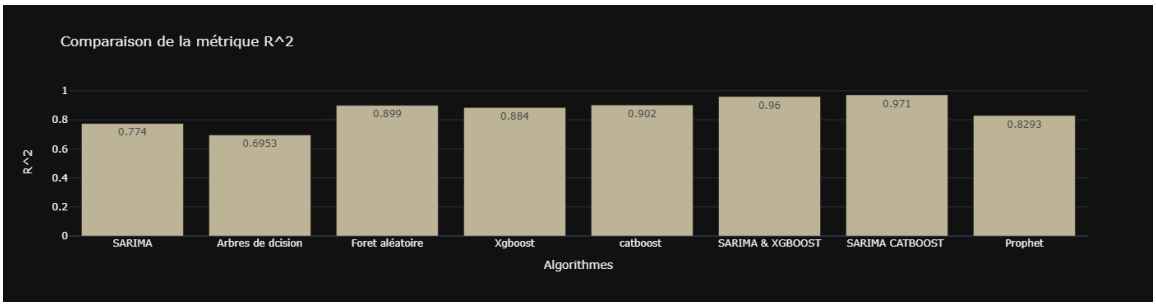

FIGURE 4.6 – Comparaison de la métrique R^2

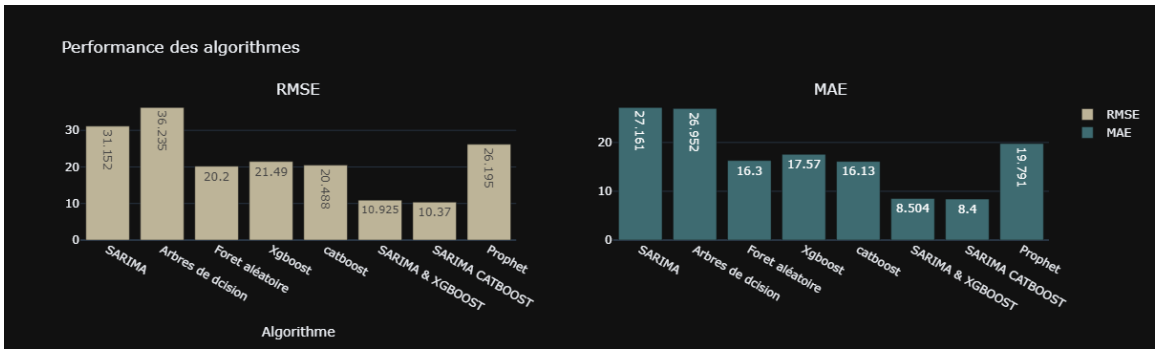

FIGURE 4.7 – Comparaison des métriques RMSE et MAE

L'analyse des différents modèles sur la série (C1,B,P3) a permis d'explorer les avantages et les limitations spécifiques à chaque modèle. Le modèle hybride SARIMA-CatBoost est le plus performant, il donne des prévisions plus précises et solides, suivi par CatBoost et forêt aléatoire qui montrent des performances similaires et fiables. Le modèle XGBoost propose une alternative avec une stabilité constante, tandis que SARIMA reste limité malgré sa capacité à capturer les tendances saisonnières. La figure 4.8 présente la comparaison des prévisions de différents modèles.

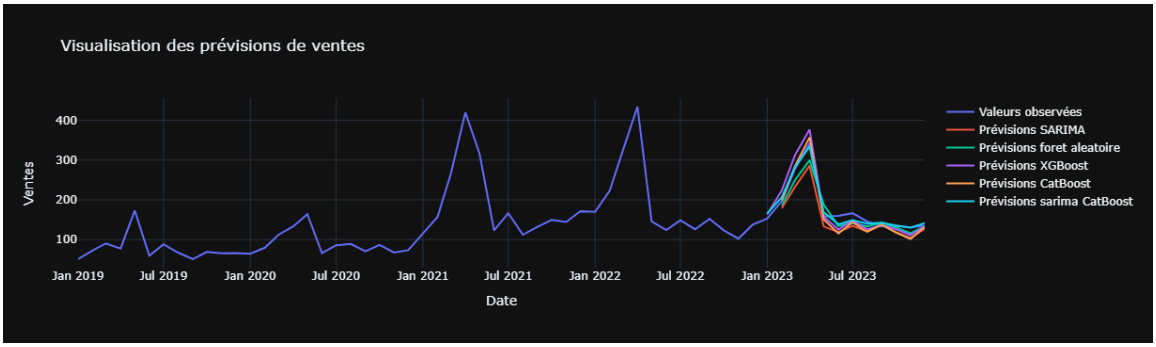

FIGURE 4.8 – Comparaison des prévisions des différents algorithmes

4.2 Évaluation de divers algorithmes sur plusieurs séries chronologiques

Hormis ce qui a été présenté précédemment, nous avons comparé les performances de différents algorithmes d'apprentissage automatique pour la prévision des ventes en évaluant plusieurs combinaisons d'hyperparamètres sur diverses séries chronologiques. Nous avons également testé d'autres algorithmes sur l'ensemble des séries chronologiques disponibles. Prophet, développé par Facebook pour les séries chronologiques avec des tendances saisonnières et des points de variation, a été inclus en raison de son approche novatrice et de son efficacité prouvée dans diverses applications.

Les résultats détaillés dans le tableau 4.9 comparent les performances des algorithmes, soulignant les meilleures combinaisons d'hyperparamètres pour chaque série, améliorant ainsi la précision des prévisions.

Algorithme	Combinaison	Hyperparametres	RMSE	MAE	R^2
SARIMA	C1-A-P1	Paramètres du modèle SARIMA : (0, 1, 1) Paramètres saisonniers du modèle SARIMA : (1, 0, 0, 12)	34.407	20.528	0.688
	C2-A-P1	Paramètres du modèle SARIMA : (0, 0, 0) Paramètres saisonniers du modèle SARIMA : (0, 1, 0, 12)	51.883	44.734	0.812
	C3-C-P3	Paramètres du modèle SARIMA : (0, 1, 1) Paramètres saisonniers du modèle SARIMA : (2, 0, 0, 12)	5.359	4.270	0.631
Arbre de décision	C1-A-P1	max_depth': 20, 'min_samples_leaf': 2, 'min_samples_split': 5	27.571	24.393	0.603
	C2-A-P1	max_depth': 5, 'min_samples_leaf': 1, 'min_samples_split': 2	46.069	39.68	0.85
	C3-C-P3	max_depth': 5, 'min_samples_leaf': 2, 'min_samples_split': 5	6.015	4.662	0.535
Forêt aléatoire	C1-A-P1	max_depth': 7, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 300	30.80	26.61	0,5
	C2-A-P1	max_depth': 5, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}	55.181	51.630	0.788
	C3-C-P3	max_depth': 7, 'max_features': 'log2', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 100	5,21	4.29	0,65
XGBoost	C1-A-P1	colsample_bytree': 1.0, 'gamma': 0, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.7	22.802	19.158	0.728
	C2-A-P1	colsample_bytree': 1.0, 'gamma': 0, 'learning_rate': 0.01, 'max_depth': 7, 'n_estimators': 300, 'subsample': 0.7	55.357	52.183	0.786
	C3-C-P3	colsample_bytree': 1.0, 'gamma': 0.1, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.7	5.881	4.582	0.555
CatBoost	C1-A-P1	border_count': 32, 'depth': 3, 'iterations': 300, 'l2_leaf_reg': 1, 'learning_rate': 0.2	9.345	6.870	0.954
	C2-A-P1	border_count': 32, 'depth': 7, 'iterations': 100, 'l2_leaf_reg': 5, 'learning_rate': 0.1	47.715	44.980	0.841
	C3-C-P3	border_count': 32, 'depth': 3, 'iterations': 100, 'l2_leaf_reg': 3, 'learning_rate': 0.1	5.766	4.398	0.572
SARIMA - XGBoost	C1-A-P1	border_count': 32, 'depth': 3, 'iterations': 300, 'l2_leaf_reg': 1, 'learning_rate': 0.2	9.345	6.870	0.954
	C2-A-P1	Paramètres du modèle SARIMA : (0, 0, 0) Paramètres saisonniers du modèle SARIMA : (0, 1, 0, 12) colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 7, 'n_estimators': 50, 'subsample': 0.7	50.130	43.375	0.825
	C3-C-P3	Paramètres du modèle SARIMA : (0, 1, 1) Paramètres saisonniers du modèle SARIMA : (2, 0, 0, 12) colsample_bytree': 0.7, 'learning_rate': 0.2, 'max_depth': 3, 'n_estimators': 100, 'subsample': 0.8	3.963	3.529	0.798
SARIMA - CatBoost	C1-A-P1	Paramètres du modèle SARIMA : (0, 1, 1) Paramètres saisonniers du modèle SARIMA : (1, 0, 0, 12) 'depth': 4, 'iterations': 300, 'learning_rate': 0.1, 'loss_function': 'RMSE'	7.645	5.988	0.969
	C2-A-P1	Paramètres du modèle SARIMA : (0, 0, 0) Paramètres saisonniers du modèle SARIMA : (0, 1, 0, 12) depth': 6, 'iterations': 300, 'learning_rate': 0.05, 'loss_function': 'RMSE'	45.04	40.619	0.858
	C3-C-P3	Paramètres du modèle SARIMA : (0, 1, 1) Paramètres saisonniers du modèle SARIMA : (2, 0, 0, 12) depth': 6, 'iterations': 100, 'learning_rate': 0.1, 'loss_function': 'RMSE'	3.547	3.264	0.838
Prophet	C1-A-P1	changepoint_prior_scale': 0.5, 'seasonality_mode': 'additive', 'seasonality_prior_scale': 0.01	43.94	32.84	-0.026
	C2-A-P1	changepoint_prior_scale': 0.01, 'seasonality_mode': 'multiplicative', 'seasonality_prior_scale': 0.1	90.762	58.759	0.4245
	C3-C-P3	changepoint_prior_scale': 0.1, 'seasonality_mode': 'multiplicative', 'seasonality_prior_scale': 0.1	4.221	3.409	0.660

FIGURE 4.9 – Tableau comparatif des performances des modèles de prévision

L'analyse des résultats montre l'importance d'ajuster spécifiquement les hyperparamètres pour chaque série chronologique, en utilisant des méthodes d'optimisation comme GridSearchCV pour des prévisions précises. Les modèles hybrides comme SARIMA et CatBoost ont surpassé les autres, soulignant l'efficacité de combiner différentes approches. Une validation croisée rigoureuse et un prétraitement des données sont essentiels pour éviter le surajustement et améliorer la précision des prévisions.

Remarque 4.1. Un R^2 négatif a été obtenu avec Prophet (-0.026) sur la série chronologique (C1,B,P3) malgré son succès sur d'autres. La meilleure adaptation des autres algorithmes à la variabilité et aux anomalies spécifiques de la série temporelle a permis d'améliorer la précision des prévisions, suggérant une possible sous-optimisation des hyperparamètres dans le cas de Prophet.

4.3 Tableau de bord

Ce tableau de bord interactif, créé avec Power BI, offre une vue d'ensemble des ventes de l'organisme d'accueil. Il permet de filtrer et analyser les données par région, canal de vente, année et produit. La figure 4.10 et 4.11 sont les feuilles n° 1 et n° 2 respectivement du tableau de bord réalisé.

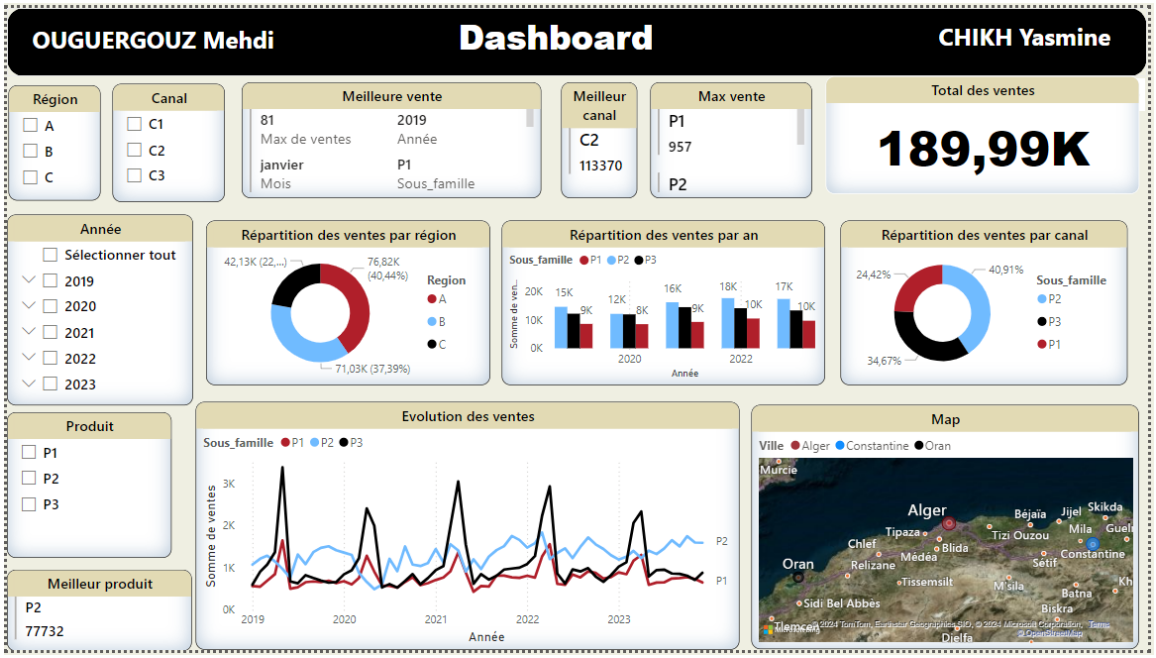

FIGURE 4.10 – Feuille n°1 du tableau de bord

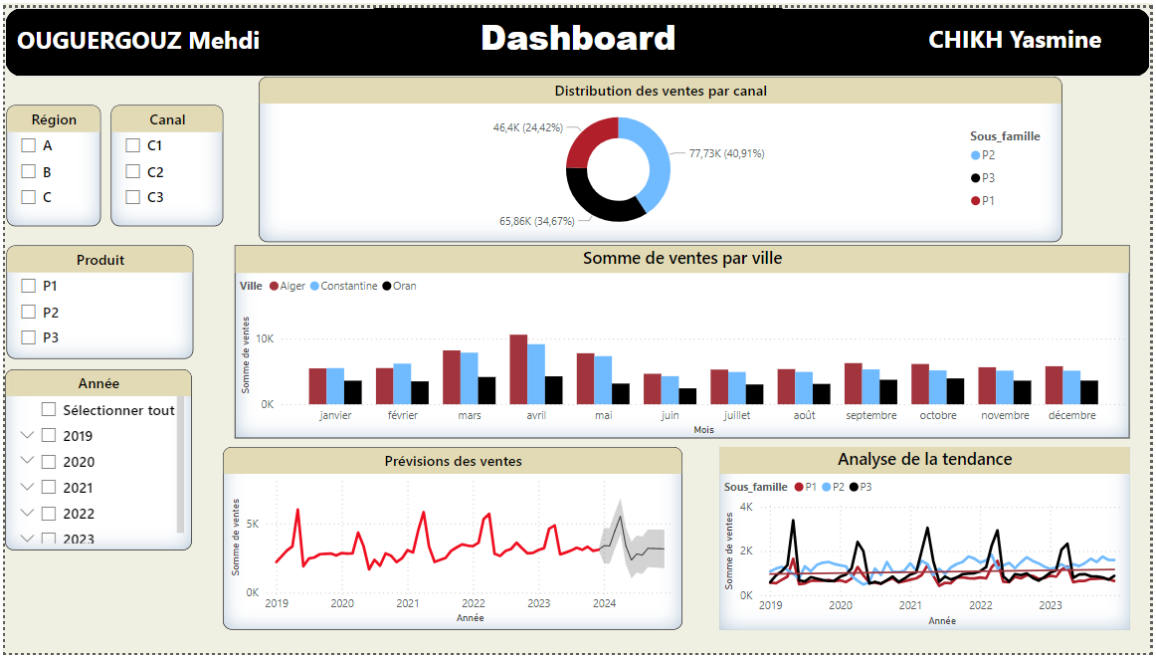

FIGURE 4.11 – Feuille n°2 du tableau de bord

Les sections principales incluent :

- Filtres de sélection : Permettent de choisir les régions, canaux, années et produits pour affiner l'analyse.
- Indicateurs clés (KPI) : Montrent la meilleure vente, le canal le plus performant, et le total des ventes.
- Visualisations : Englobent
  - Répartition des ventes par région et par canal : Diagrammes circulaires permettant de visualiser la part des ventes par différentes régions et canaux.
  - Répartition des ventes par année : Histogramme détaillant les ventes annuelles, permettant de comparer les performances de vente au fil des années.
  - Somme des ventes par ville : Histogrammes comparant les sommes des ventes dans différentes villes pour chaque mois de l'année.
  - Évolution des ventes : Graphiques linéaires montrant les tendances des ventes sur plusieurs années, permettant d'identifier les périodes de forte et faible activité.
  - Prévisions des ventes : Un graphique linéaire avec une zone d'incertitude illustrant les prévisions de ventes futures basées sur les données historiques.
  - Analyse de la tendance : Un graphique linéaire détaillant les tendances de vente par sous-famille de produits, aidant à comprendre quelles catégories de produits sont en croissance ou en déclin.
  - Localisation des ventes principales : Carte géographique interactive indiquant les villes avec les ventes les plus significatives, aidant à visualiser les zones géographiques de forte activité.

Le tableau de bord facilite l'analyse des performances de vente et aide à prendre des décisions informées, nous permettons de suivre les performances de vente en temps réel, de repérer

les tendances et d'identifier les opportunités de croissance. Les capacités de filtrage et d'interaction offrent une flexibilité accrue pour analyser les données sous différents angles et niveaux de détail.

Conclusion

Dans ce chapitre, nous avons discuté des résultats obtenus à partir des approches étudiées de modélisation et de prévision des ventes. Nous avons examiné les performances de plusieurs modèles tels que SARIMA, forêt aléatoire, XGBoost, CatBoost, ainsi que le modèle hybride SARIMA-CatBoost que nous avons proposé.

Après avoir procédé à une analyse comparative des différents modèles, les résultats attestent la supériorité du modèle hybride SARIMA-CatBoost, qui tire parti des points forts des techniques classiques et des méthodes d'apprentissage automatique. Ce modèle a montré une amélioration significative en termes de précision des prévisions, validée par des métriques d'évaluation telles qu'une MAE de 8.40, un RMSE de 10.37 et un R^2 de 97%. Cela a souligné l'importance cruciale d'une approche adaptative et flexible dans le domaine de la prévision des ventes, illustrant ainsi la pertinence et l'efficacité des méthodes employées dans des conditions réelles.

Conclusion générale

Ce mémoire a pour objet d'étudier l'application des techniques d'analyse prédictive et d'apprentissage automatique au sein de CEVITAL Agro-Alimentaire, avec un accent particulier sur les méthodes de prévision des séries chronologiques. À travers les quatre chapitres, nous avons exploré différentes facettes de cette intégration, en commençant par une introduction à l'entreprise et aux concepts de base des séries chronologiques, puis en approfondissant l'apprentissage automatique.

Notre étude a débuté par l'application de modèles classiques tels que SARIMA, qui sont bien établis pour leur capacité à capturer les composantes saisonnières des séries chronologiques. Cependant, pour améliorer davantage la précision des prévisions, nous avons exploré divers algorithmes d'apprentissage automatique, notamment les forêts aléatoires, CatBoost et XGBoost qui ont montré une amélioration significative de la précision des prévisions. La combinaison de ces modèles d'apprentissage automatique avec des modèles classiques comme SARIMA, aboutissant à l'approche hybride, a montré des résultats encore plus prometteurs. L'hybridation des modèles a permis de capturer à la fois les tendances saisonnières et les non-linéarités présentes dans les données. Cette approche offre un avantage compétitif permettant à l'entreprise de mieux anticiper la demande et de gérer les ressources de manière plus efficace.

La performance des modèles hybrides a été évaluée à l'aide de métriques telles que RMSE, MAE et R^2, démontrant leur supériorité par rapport aux modèles classiques seuls. Les résultats de la phase de test montrent une réduction significative des erreurs de prévision, ce qui indique que les modèles proposés sont robustes et fiables. En outre, les études de cas présentées ont illustré des applications concrètes, confirmant la pertinence des méthodes employées dans des scénarios réels.

Toutefois, cette étude comporte quelques limites. Pour certaines périodes et catégories de produits, le manque de données historiques a restreint la capacité des modèles à détecter des tendances et des schémas complexes, ce qui a eu un impact sur la précision des prévisions. De plus, l'indisponibilité et la qualité parfois insuffisante des données utilisées pour l'entraînement des modèles restent des préoccupations majeures. Des lacunes dans la collecte, la gestion ou la disponibilité des données peuvent entraîner des prévisions inexactes, soulignant ainsi la nécessité de données précises et en temps réel comme un défi constant.

L'impact de la pandémie de COVID-19 a également été pris en compte dans notre analyse. Les données historiques incluent désormais les années perturbées par la pandémie, introduisant des variations inattendues dans les comportements d'achat et les tendances du marché. Ces fluctuations rendent la projection des tendances futures plus complexe et incertaine, ce qui souligne l'importance d'adapter les modèles pour être plus résistants aux perturbations externes.

L'étude ouvre plusieurs perspectives intéressantes pour des travaux futurs. Voici quelques-unes des directions prometteuses pour approfondir et étendre cette recherche :

- **Incorporation de facteurs externes :** Intégrer des indicateurs économiques ou avis des clients provenant des réseaux sociaux pour enrichir l'analyse des sentiments par le traitement du langage naturel (NLP). Ces informations seront ensuite stockées dans des datacenters.
- **Technologies émergentes :** Utiliser des techniques de pointe comme les réseaux de neurones récurrents (RNN) et les réseaux de neurones à mémoire à long terme (LSTM) pour capturer des dynamiques temporelles complexes.
- **Amélioration des modèles hybrides :** Il serait bénéfique d'explorer d'autres combinaisons d'autres algorithmes. La synergie entre eux pourrait permettre de tirer parti des forces de chaque type de modèle.
- **IoT et données en temps réel :** L'IoT offre une visibilité accrue et une réactivité améliorée grâce à la collecte de données en temps réel et au suivi précis des stocks. Ces dispositifs permettent également une maintenance prédictive et une optimisation de la logistique. L'analyse des données IoT améliore les prévisions et aide les entreprises à ajuster rapidement leurs stratégies en réponse aux tendances du marché, renforçant ainsi leur compétitivité et leur efficacité opérationnelle.
- **TimeGPT** : TimeGPT de Nixtla, en partenariat avec OpenAI, ouvre de nouvelles perspectives pour les prévisions de ventes. Cette technologie combine l'expertise de Nixtla dans la modélisation temporelle avec les capacités avancées d'OpenAI en génération de texte. En intégrant ces innovations, les entreprises peuvent bénéficier de prévisions plus précises et adaptatives, facilitant ainsi une gestion proactive des inventaires et une optimisation des stratégies commerciales en réponse aux dynamiques du marché.

Ainsi, ce mémoire n'est pas seulement une exploration académique, mais une réponse pratique et stratégique aux défis contemporains de la prévision des ventes, avec l'ambition de positionner Cevital SPA à la pointe de l'innovation technologique et de la compétitivité. Ce travail offre une base pour la prise de décision et une gestion proactive des ressources. L'implémentation de ces méthodes avancées permettra non seulement d'améliorer les prévisions de ventes, mais aussi de renforcer la réactivité et l'agilité de l'entreprise face aux dynamiques changeantes du marché. Les perspectives d'avenir, telles que l'intégration de nouvelles sources de données, l'utilisation de modèles encore plus sophistiqués et la mise en place de systèmes de prévision en temps réel, offriront à Cevital SPA les outils nécessaires pour maintenir et accroître son avantage concurrentiel, assurant ainsi une croissance durable et stratégique dans un environnement économique en constante évolution.

Bibliographie

[1] S. Agomoh & I. Ukabuiro – « Intelligent sales forecasting technique application », *European Journal of Theoretical and Applied Sciences* **1** (2023), no. 6, p. 641–653.

[2] Y. Ahmadov & P. Helo – « Deep learning-based approach for forecasting intermittent online sales », *Discover Artificial Intelligence* **3** (2023), p. 45.

[3] C. A. Azencott – *Introduction au machine learning*, 2e éd., Dunod, 2022.

[4] L. Breiman – *Forêts aléatoires*, vol. 45, Apprentissage automatique Publishers, 2001.

[5] P. J. Brockwell & R. A. Davis – *Time series : Theory and methods*, 2e éd., Springer-Verlag, New York, 1996.

[6] O. Celik – « A research on machine learning methods and its applications », *Journal of Educational Technology and Online Learning* **1** (2018), no. 3, p. 25–40.

[7] T. Chen & C. Guestrin – « Xgboost : Un système de boosting d'arbres évolutif », in *Actes de la 22e conférence internationale ACM SIGKDD sur la découverte des connaissances et l'exploration de données*, août 2016, p. 785–794.

[8] S. Cheriyan, S. Ibrahim, S. Mohanan & S. Treesa – « Intelligent sales prediction using machine learning techniques », in *Proceedings of the International Conference on Computing, Electronics & Communications Engineering (iCCECE)* (Muscat, Sultanate of Oman), August 2018, p. 53–58.

[9] Y. Chirouze – *Prévoir ses ventes*, Edition Chotard & Associés, Paris, 1986, Mémoire de maîtrise.

[10] J. S. A. (Ed.) – *Principles of forecasting : A handbook for researchers and practitioners*, vol. 30, Kluwer Academic, Boston, MA, 2001.

[11] Y. Freund & R. E. Schapire – « A decision-theoretic generalization of on-line learning and an application to boosting », *Journal of Computer and System Sciences* **55** (1997), no. 1, p. 119–139.

[12] R. Garnier – « Machine apprentissage sur les séries temporelles et applications à la prévision des ventes pour l'e-commerce », Thèse de doctorat, CY Cergy Paris Université, 2021, NNT : 2021CYUN1051.

[13] C. Gouriéroux & A. Monfort – *Séries temporelles et modèles dynamiques*, FeniXX, 1995.

[14] A. Géron – *Apprentissage automatique pratique avec scikit-learn, keras et tensorflow*, O'Reilly Media, 2022.

[15] R. J. Hyndman & G. Athanasopoulos – *Forecasting : principles and practice*, 2e éd., OTexts, Melbourne, Australia, 2018.

[16] S. Jabeur, C. Gharib, S. Mefteh-Wali & W. Arfi – « Modèle catboost et techniques d'intelligence artificielle pour la prédiction des défaillances d'entreprise », *Prévision technologique et changement social* **166** (2021), p. 120658.

[17] W. McKinney – *Python for data analysis*, O'Reilly Media, Inc., 2022.

[18] Microsoft – « An introduction to power bi », s.d., Consulté le 14 juin 2024.

[19] M. Negnevitsky – *Artificial intelligence : a guide to intelligent systems*, Pearson education, 2005.

[20] S. Nene – « Deep learning for natural language processing », *International Research Journal of Engineering and Technology (IRJET)* **04** (2017), no. 11, p. 933, Dept. of Computer Engineering, VESIT, Maharashtra, India.

[21] Y.-S. Shih & M.-H. Lin – « A istm approach for sales forecasting of goods with short-term demands in e-commerce », in *Proceedings of the 11th Asian Conference on Intelligent Information and Database Systems (ACIIDS)*, 2019, p. 244–256.

[22] A. Tack, T. François, A. L. Ligozat & C. Fairon – « Modèles adaptatifs pour prédire automatiquement la compétence lexicale d'un apprenant de français langue étrangère », *JEP-TALN-RECITAL 2016* **2** (2016), p. 221–234.

[23] M. Ulrich, H. Jahnke, R. Langrock, R. Pesch & R. Senge – « Sélection de modèles basée sur la classification dans la prévision de la demande de détail », *Revue internationale de prévision* **38** (2022), no. 1, p. 209–223.

[24] C. Willmott & K. Matsuura – « Avantages de l'erreur absolue moyenne (mae) par rapport à l'erreur quadratique moyenne (RMSE) dans l'évaluation des performances moyennes du modèle », *Recherche climatique* **30** (2005), no. 1, p. 79–82.

Résumé

Ce mémoire se concentre sur l'amélioration des techniques de prévision des ventes appliquées à Cevital SPA. L'objectif principal est d'améliorer la précision des prévisions des ventes en comparant différents algorithmes de modélisation des séries chronologiques. La démarche adoptée inclut une analyse des séries chronologiques de ventes, l'intégration et l'entraînement de modèles prédictifs classiques et d'apprentissage automatique tels que SARIMA, les arbres de décision, les forêts aléatoires, XGBoost, CatBoost, et Prophet. En se basant sur différentes métriques (RMSE, MAE, R^2) nous avons évalué la précision des modèles et comparé leurs performances respectives, puis démontré les avantages apportés par l'approche hybride.

Mots-clés : Prévision des ventes, Séries chronologiques, Apprentissage automatique, SARIMA, Arbres de décision, Forêts aléatoires, XGBoost, CatBoost, Prophet, Approce hybride.

Abstract

This thesis focuses on improving sales forecasting techniques applied to Cevital SPA. The primary objective is to enhance the accuracy of sales forecasts by comparing different time series modeling algorithms. The approach includes an analysis of sales time series, the integration and training of classical predictive models and machine learning models such as SARIMA, decision trees, random forests, XGBoost, CatBoost, and Prophet. Using various metrics (RMSE, MAE, R^2), we evaluated the accuracy of these models and compared their respective performances, highlighting the advantages brought by the hybrid approach.

Keywords : Sales forecasting, Time series, Machine learning, SARIMA, Decision trees, Random forests, XGBoost, CatBoost, Prophet, Hybrid approach.