A propos de la réalisation de ce test

Afin de répondre aux questions de ce test technique, le travail a été réalisé de la façon suivante :

- Analyse des données et traitement de ces dernières.
- Modélisation des prédictions des ventes futures. Pour tester ce modèle, nous avons scindé le jeu de données de la façon suivante : du 1er janvier 2014 au 31 décembre 2016, les données sont utilisées pour l'entraînement. L'année 2017 est utilisée comme jeu de test.
- Les horizons de prédiction sont de 30, 90 et 365 jours.
- Le code a été écrit en programmation orientée objet pour plus de modularité et est stocké dans le dossier scripts.
- Toutes les étapes ont été présentées dans le notebook *application_code.ipynb*.

Présentation sommaire du travail effectué

L'objectif était de prédire les ventes futures pour des gestionnaires de magasins. Nous étions donc face à un problème de séries temporelles. Afin d'utiliser au mieux le Machine Learning pour ces prévisions, nous avons d'abord étudié la structure des données lors d'une phase d'analyse exploratoire (**EDA**), puis nous avons traité les données jugées peu pertinentes ou erronées lors d'une phase de **data cleaning**. Cette étape a été suivie d'un **feature engineering**, visant à améliorer la capacité prédictive de notre modèle.

Pour la modélisation, nous avons utilisé deux approches :

- Un modèle **ARIMA** comme baseline.
- Un modèle de **Machine Learning** non paramétrique, **XGBoost**.

Nous avons choisi la **RMSE** comme métrique à minimiser afin d'évaluer la performance de nos modèles et de les comparer.

Question #1 : Préparation des données

- Si tu rencontres des problèmes de qualité des données durant ta manipulation des données de ventes, comment les as-tu résolus?

Réponse :

Lors de la réception des données, le premier problème rencontré fut l'encodage du fichier CSV. Ce dernier, étant au format MacRoman, a dû être converti en UTF-8, un format plus universel et utilisé par défaut pour l'ouverture des fichiers CSV avec la librairie Pandas.

À la suite d'une première analyse des données, j'ai pu constater que celles-ci étaient de plutôt bonne qualité (aucune valeur manquante, pas de doublons purs dans les données), bien que quelques ajustements aient été nécessaires :

- Les variables de date (*Order Date* et *Ship Date*) n'étaient pas au format datetime, comme il est d'usage, mais au format object. Un reformatage a donc été appliqué.
- Concernant l'étude des données dupliquées, nous nous sommes penchés sur les doublons liés aux IDs. J'ai constaté qu'un consommateur avait commandé le même produit, le même jour, avec le même mode de livraison, mais à des quantités différentes, entraînant une variation du profit et des ventes. Afin de ne pas fausser les données lors de l'apprentissage,

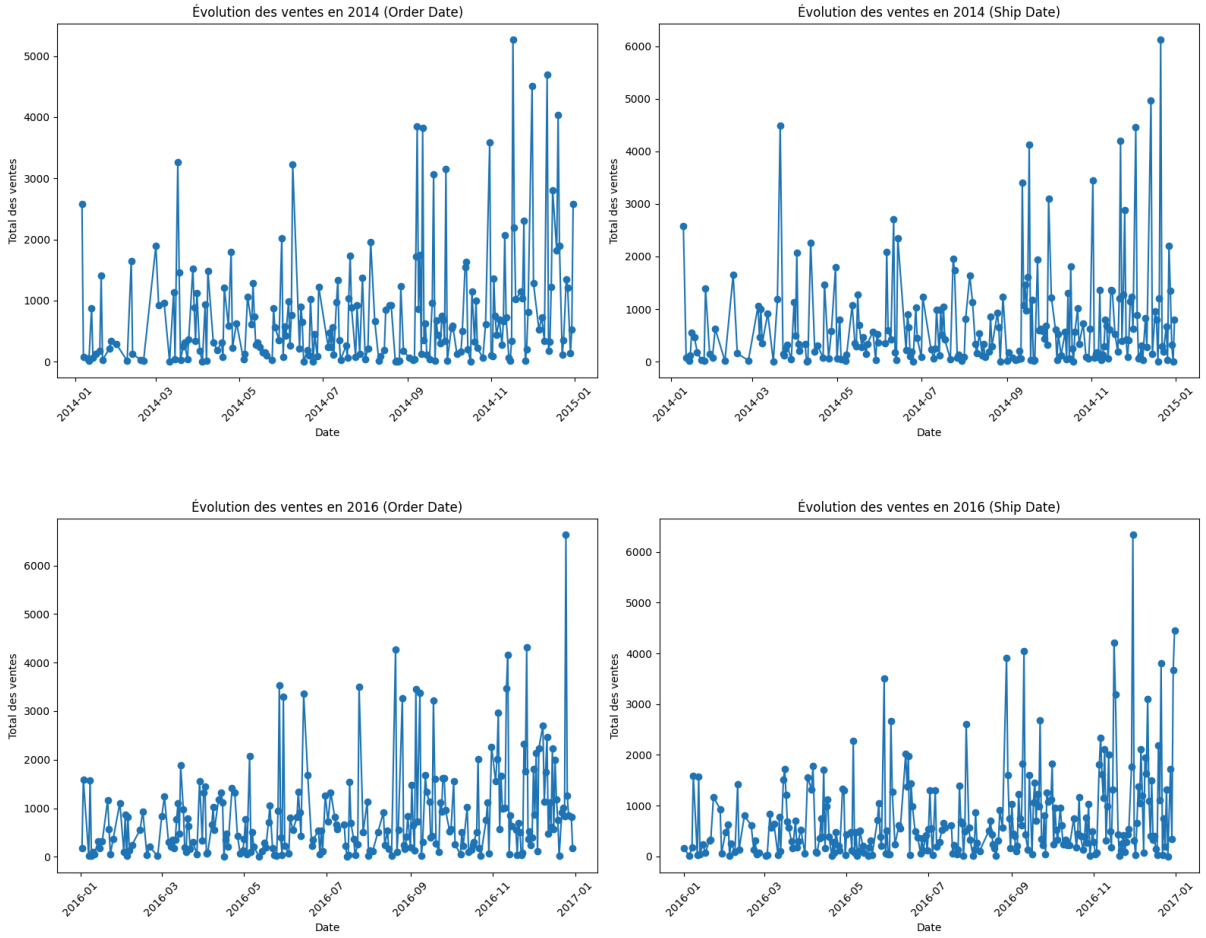
j'ai décidé de regrouper ces deux lignes, considérant qu'il s'agissait d'une commande fragmentée.

- Certaines variables (*State* et *Category*) ne comportaient qu'une seule catégorie. N'apportant pas d'intérêt à notre étude, j'ai décidé de m'en séparer.
- Lors de l'analyse de la distribution des variables quantitatives, des valeurs fortement négatives (inférieures à -1000) dans la variable Profit ont attiré mon attention, déformant le caractère gaussien de sa distribution. Considérant qu'il s'agissait de valeurs aberrantes, je les ai retirées du dataset, ce qui a permis de retrouver une distribution plus gaussienne pour Profit.

- Limite-toi aux trois enjeux les plus pertinents selon toi (appuie-toi avec un visuel).

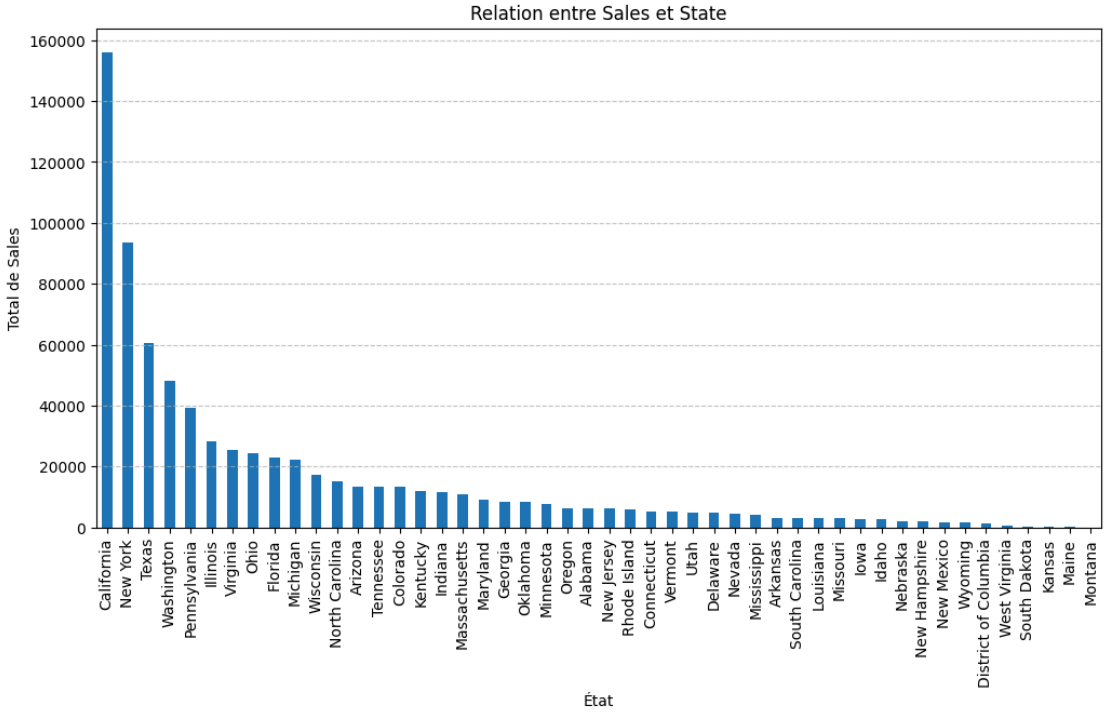
Réponse : Lors de l'analyse des données, j'ai pu me rendre compte de trois points clés dans nos données qui auront un impact sur la phase de data engineering et de modélisation :

1. Lorsque j'ai analysé l'évolution des ventes sur l'ensemble du dataset, j'ai détecté une forme de saisonnalité : la majorité des ventes avaient lieu en fin d'année (vers l'hiver). Cette tendance s'est confirmée en observant l'évolution des ventes année après année, comme on peut le constater sur les graphiques suivants :



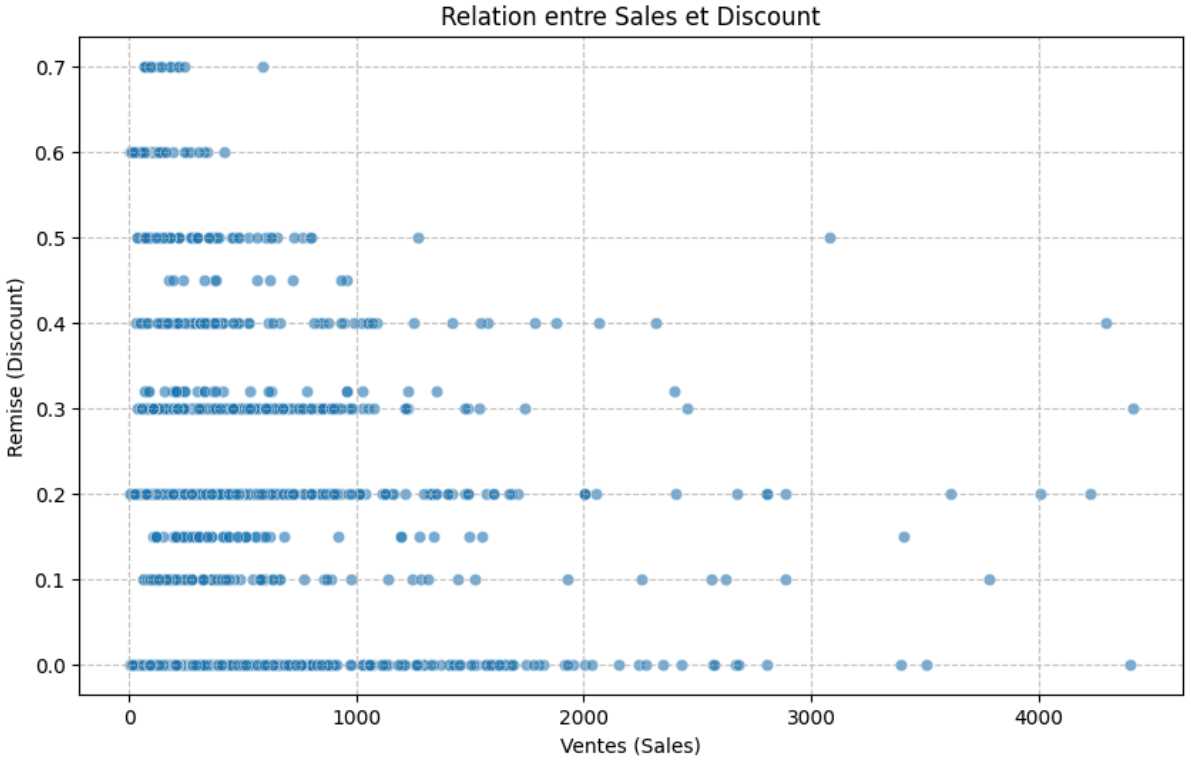
Un enjeu majeur ici sera de trouver un moyen pour que le modèle intègre ce caractère saisonnier des ventes afin de les répliquer au mieux dans le futur.

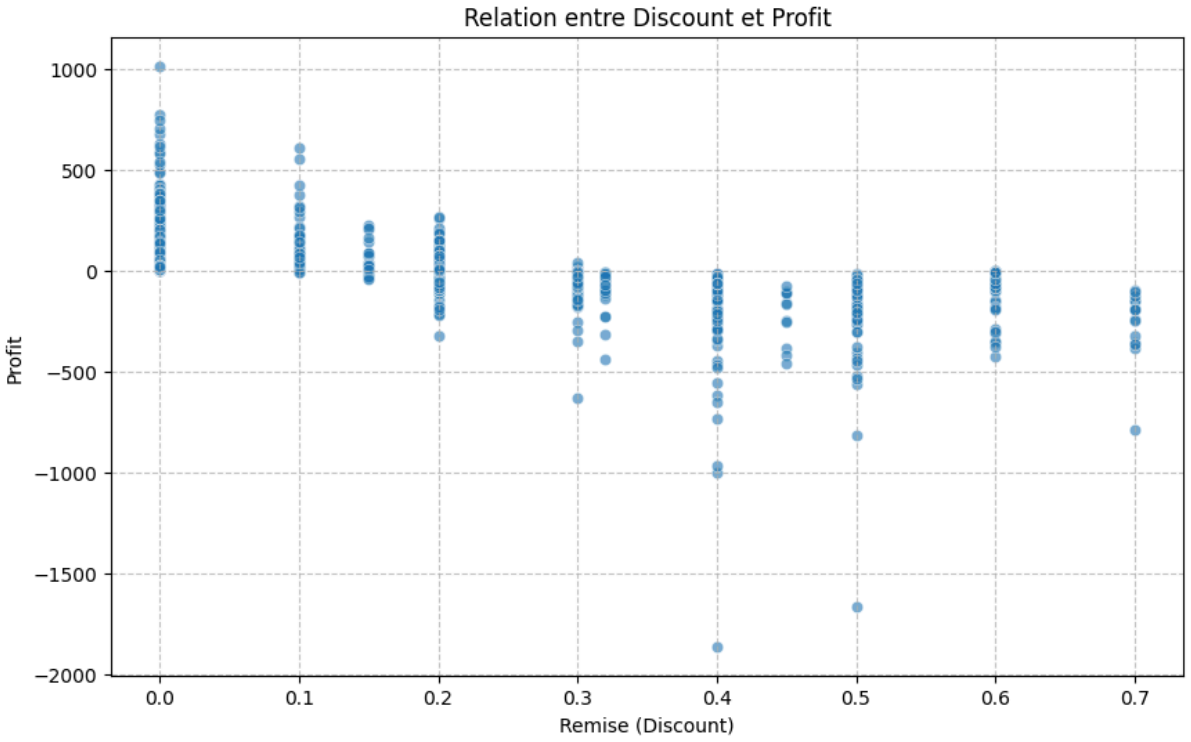
2. Un second enjeu ici sera de gérer les déséquilibres de classe entre certaines catégories. Par exemple, en ce qui concerne la répartition des ventes entre les États, il existe une différence flagrante entre les 42 États. De plus, la gestion de cette haute cardinalité devra également être prise en compte afin d'éviter le problème du "curse of dimensionality".



3. Enfin, un autre enjeu sera de prendre en compte la relation entre les réductions et les ventes réalisées. Normalement, plus un produit bénéficie d'un rabais important, plus ses ventes devraient augmenter. Or, comme le montrent les graphiques suivants, au-delà d'un certain taux de remise, les ventes ralentissent, impactant ainsi les profits (par exemple, avec une

remise de 70 %)





- Est-ce que les insights trouvés peuvent être transformés en features qui faciliteront l'apprentissage du modèle ML?

Réponse : Oui, ceci est tout à fait possible :

- Concernant la saisonnalité, nous avons scindé les dates en quatre catégories : hiver (*décembre, janvier, février*), printemps (*mars, avril, mai*), été (*juin, juillet, août*) et automne (*septembre, octobre, novembre*). Ces catégories ont ensuite été one-hot encodées afin d'être interprétables par notre modèle.
- Pour les États, la stratégie du target encoding a été choisie en raison du grand nombre de catégories associées à cette variable. Afin d'éviter le data leakage, nous avons utilisé un leave-one-out target encoding, qui consiste à remplacer chaque valeur de *State* par la moyenne des *Sales* pour cette catégorie, en excluant la ligne courante.
- Des variables temporelles ont également été créées, telles que le mois de commande, le jour où elle a été passée, ainsi que le délai entre la date de commande et la date d'expédition.
- Enfin, la variable Sales a été laggée de 30 jours, ce qui permet de capturer les tendances saisonnières.

Question #2 : Insights et interprétation

- En tenant compte des parties prenantes visées par ta solution, comment interprètes-tu les résultats produits par ta solution ML? Comment cette solution ajoute-t-elle de la valeur pour ces parties prenantes?

Réponse: Les résultats produits par la solution de Machine Learning développée mettent en évidence plusieurs points :

- Une prévision à court terme (un mois) est plus précise qu'une prévision à long terme. Les parties prenantes doivent donc concentrer leurs décisions sur cette période afin d'optimiser leurs stratégies de vente.
- Certaines features influencent fortement les ventes, un facteur à prendre en compte par les parties prenantes (par exemple, l'impact des ventes de certaines sous-catégories de produits au cours de l'année).
- Avec un modèle simple, les premiers résultats sont prometteurs, ce qui montre qu'il est possible d'améliorer encore nos performances et d'obtenir des prévisions très précises. Cela pourrait être envisagé en combinant plusieurs modèles, qu'ils soient issus du ML ou du Deep Learning, comme un LSTM.
- Cette solution apporte de la valeur aux parties prenantes, en leur permettant de mieux comprendre leurs ventes, de mieux cibler leur clientèle, et d'anticiper efficacement les ventes futures.

- Selon toi, comment envisages-tu que les parties prenantes utilisent ta solution pour tenter de comprendre comment augmenter les ventes?

Réponse: Ils peuvent utiliser ces résultats de différentes manières :

1. L'analyse des données et l'extraction des insights peuvent être exploitées pour mieux adapter leur stratégie marketing, ce qui permettrait d'augmenter les ventes futures et/ou d'ajuster leur approche. Par exemple, ils pourraient décider de ne plus appliquer de remises supérieures à un certain seuil si cela entraîne une baisse significative du profit.
2. L'utilisation de la solution ML, et en particulier l'analyse de l'importance de chaque variable dans le modèle, peut servir d'aide à la prise de décision quant à l'évolution des ventes futures. Cela permettrait d'optimiser les ventes en s'appuyant sur les facteurs ayant le plus d'impact.

Question #3 : Solution ML

Pour réaliser la modélisation des ventes, j'ai d'abord scindé mon jeu de données en train et test. Étant donné qu'il s'agit d'un problème de séries temporelles, j'ai effectué un split temporel, avec les données jusqu'à fin 2016 pour l'entraînement et celles de 2017 pour le test.

La métrique choisie est la RMSE, car elle est interprétable et moins sensible aux valeurs aberrantes que la MSE (bien que ces dernières aient été traitées dans la partie data cleaning, il est préférable d'adopter une approche conservatrice sur ce point).

Nous comparerons donc les ventes prédites pour 2017 par le modèle aux ventes réelles.

Modélisation d'un modèle baseline (ARIMA)

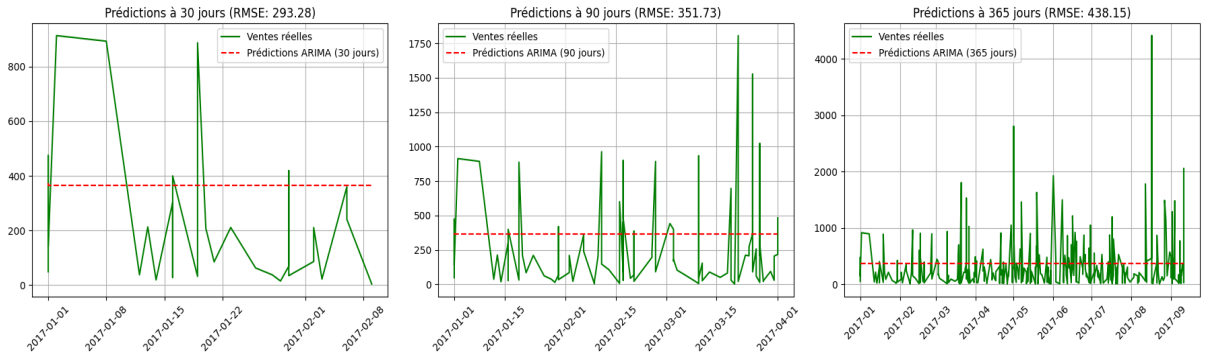
Comme dans toute mission de time series, j'ai choisi dans un premier temps un modèle baseline afin de comparer ses performances à celles du modèle de Machine Learning. Le modèle baseline retenu est un ARIMA(1,1,1), une approche très basique pour notre cas d'usage.

J'ai également incorporé une fenêtre glissante dans les prévisions du modèle, lui permettant d'apprendre des prévisions faites au pas précédent.

Enfin, nous avons réalisé des prévisions à trois horizons : 30, 90 et 365 jours.

Résultats de la modélisation

Comme prévu, le modèle ARIMA affiche des performances médiocres : il n'intègre aucune tendance et ne parvient pas à détecter les évolutions des prix, comme on peut le constater sur le graphique suivant :



Notre modèle de Machine Learning est un XGBoost. Ce dernier a l'avantage de s'appuyer sur des arbres de décision, mais avec la particularité d'estimer des weak learners (de très petits arbres, juste un peu meilleurs que l'aléatoire). Chaque arbre prend en compte les erreurs du précédent de manière itérative, ce qui améliore progressivement les performances du modèle.

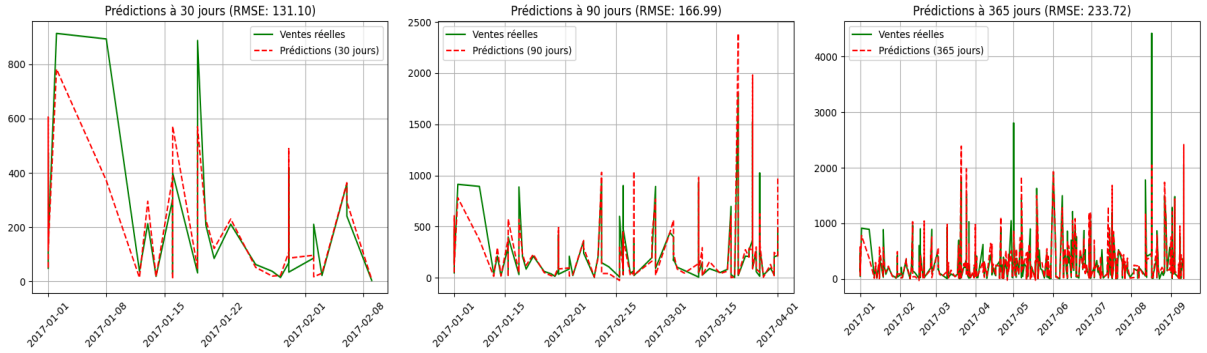
J'ai choisi XGBoost pour son architecture en level-wise growth, qui a moins tendance à sur-apprendre et donc à mal généraliser.

Optimisation des hyperparamètres

Pour l'hyperparamétrage, j'ai utilisé la librairie Optuna, qui m'a permis de minimiser au mieux la RMSE et d'optimiser les hyperparamètres sur 50 itérations. Pour des raisons de temps d'exécution, ces hyperparamètres ont ensuite été fixés manuellement dans le code.

Résultats

Comme prévu, les résultats sont nettement meilleurs que ceux d'ARIMA, bien qu'ils restent imparfaits. Toutefois, le modèle montre une capacité à capturer une tendance saisonnière, principalement sur 30 et 90 jours, comme on peut l'observer sur le graphique suivant.



Ce modèle présente un inconvénient majeur : en cas de volume de données élevé, il peut être très long à exécuter, rendant l'utilisation d'un GPU nécessaire, ce qui peut s'avérer coûteux. De plus, un hyperparamétrage trop poussé peut entraîner du sur-apprentissage, réduisant ainsi les performances du modèle lors de sa mise en production.

La RMSE de notre modèle reste encore trop élevée, mais elle est optimisable. Un travail plus approfondi sur le feature engineering, un encodage plus adapté des variables catégorielles, ainsi que l'intégration de davantage de données constituent des pistes d'amélioration pour renforcer les performances du modèle.

Question #4 : Dégradation de la performance

Plusieurs facteurs peuvent expliquer la dégradation des performances de mon modèle lors de sa mise en production. L'un d'eux pourrait être un choc exogène, inconnu dans l'historique des données du modèle, perturbant totalement les ventes (par exemple, une crise économique, une pandémie mondiale, etc.). Ce phénomène est plus communément connu sous le nom de **drift** des données.

Une solution pour y remédier serait de réentraîner le modèle avec des données plus récentes, intégrant ainsi ce choc dans les données d'entraînement, ce qui permettrait au modèle de mieux prendre en compte ces évolutions. Une autre approche consisterait à améliorer la fenêtre glissante déjà mise en place dans le code, afin que le modèle s'entraîne en continu sur de nouvelles données.

Question #5 : Intégration de l'IA générative

- Donne un exemple d'architecture où l'IA générative pourrait être utilisée.

Dans notre cas, l'IA générative pourrait être utilisée pour aider les gestionnaires de magasins à comprendre l'évolution de leurs ventes et à prédire leurs tendances futures.

En effet, la mise en place d'un RAG (utilisant des LLM et l'architecture Transformer) capable d'exploiter les données de ventes des magasins permettrait aux gestionnaires d'analyser les principaux facteurs influençant leurs performances.

Ce RAG pourrait également fournir des KPIs pertinents, comme l'impact des remises trop élevées sur les ventes et les profits, afin d'éclairer les décisions stratégiques.

De plus, ce RAG pourrait réaliser une première modélisation des ventes à venir, offrant ainsi aux gestionnaires un indicateur fiable pour les aider à anticiper et ajuster leurs décisions.

- Propose un exemple de prompt qui pourrait être utilisé pour interagir avec l'IA générative.

Un prompt pour interagir avec l'IA générative peut être le suivant :

"Je suis gestionnaire d'une entreprise qui vend des fournitures pour la maison. Mon objectif est d'améliorer mes résultats et d'anticiper mes ventes futures.

Dans un premier temps, je veux que tu analyses les données que je t'ai fournies et que tu me transmettes des graphiques et KPIs clés pour mieux comprendre mes ventes. Je souhaite notamment visualiser :

- Le profil de mes clients (type de client, fréquence d'achat, panier moyen).
- Les périodes de l'année les plus prolifiques en termes de ventes.
- L'impact des soldes et des remises sur les ventes et les profits.

Ensuite, je veux que tu prévoies mes ventes pour les 6 prochains mois en expliquant ta démarche. Tu utiliseras un algorithme de Machine Learning ou de Deep Learning adapté aux données temporelles. Assure-toi d'expliquer les variables les plus influentes et de présenter les résultats sous forme de graphiques et tableaux interprétables."