Rapport Projet Modèles de prévision et Deep Learning
Avril 2023

Rapport

Projet Modèles de prévision et Deep Learning

Étudiants :
Giulia DALL'OMO
Amande EDO

Enseignants :
Guillaume Houchard

Table des matières

Table des figures

1 Introduction

Dans le cadre du cours de Modèles de prévisions et Deep Learning enseigné par Mr Guillaume Hochard, nous avons ainsi choisi de réaliser **un modèle de prévision des quantités journalières vendues au sein d'une boulangerie française**. En ce sens notre problème de machine learning est :un enjeu de **forecast des quantités journalières vendues** au sein de la boulangerie.

2 Présentation du jeu de données

Notre jeu de données provient d' **une boulangerie française**, elle contient des informations sur les **transactions quotidiennes effectuées par les clients sur une période de près de deux ans**, débutant le 1er janvier 2021 et se terminant le 30 septembre 2022.

Plus spécifiquement, le jeu de données contient **234005 observations**, plus de 136000 transactions et **6 colonnes** qui sont respectivement :

- date : date de la transaction
- time : heure de la transaction
- ticket number : ticket d'identification unique de la transaction
- article : nom de l'article acheté
- quantity : quantité acheté
- unit price : prix unitaire par article

3 Analyse exploratoire des données

	Unnamed: 0	date	time	ticket_number	article	Quantity	unit_price
0	0	02/01/2021	08:38	150040.0	BAGUETTE	1.0	0,90 €
1	1	02/01/2021	08:38	150040.0	PAIN AU CHOCOLAT	3.0	1,20 €
2	4	02/01/2021	09:14	150041.0	PAIN AU CHOCOLAT	2.0	1,20 €
3	5	02/01/2021	09:14	150041.0	PAIN	1.0	1,15 €
4	8	02/01/2021	09:25	150042.0	TRADITIONAL BAGUETTE	5.0	1,20 €

FIGURE 1 – Affichage de la base de données

Notre base de données contient : 234005 observations et 6 variables.

Après avoir affiché la base de données, nous avons supprimé la colonne "Unnamed : 0" car elle ne contient pas des informations qui vont nous servir dans la suite du projet.

De plus, pour harmoniser la base, nous avons transformé le nom des colonnes en miniscule.

3.1 Data cleaning

3.1.1 Valeurs manquantes

Nous vérifions dans un premier temps, si notre base de données contient des **valeurs manquantes** :

```
# Vérification de l'existence des valeurs manquantes
print(f"Notre base de données contient : " + str(df_bakery.isnull().sum().sum()) + " valeurs manquantes.")
```

```
Notre base de données contient : 0 valeurs manquantes.
```

FIGURE 2 – Proportion de valeurs manquante dans la base de données

On peut voir à travers la figure ci-dessus que notre base de données ne contient aucune valeurs manquantes donc il n'y aura pas nécéssité de traitement.

3.1.2 Doublons

Dans un second temps, nous vérifions la présence de **doublons** dans notre base de données car ceux-ci pourrait biaiser nos prédictions.

```
# Vérification de l'existence de doublons dans la base de données
print(f"Notre base de données contient : " + str(df_bakery.duplicated().sum().sum()) + " doublons.")
```

```
Notre base de données contient : 1210 doublons.
```

FIGURE 3 – Proportion de valeurs manquante dans la base de données

On peut constater que notre base de données contient des doublons, nous allons donc les supprimer afin d'avoir un jeu de données complet et sans biais.

3.1.3 Traitement des quantités négatives | Outliers

Enfin, on constate que **notre base de données contient des quantités vendues négatives**, cela pourrait correspondre à deux cas de figures :

— Une remise effectuée par la boulangerie
— Une erreur de saisie au niveau de l'encaissement

Faute d'assez de connaissance précise sur le contexte de ces quantités vendues négatives, nous allons procéder à la suppression de tous les tickets de caisses contenant une quantité vendue négative.

	quantity	unit_price
count	231506.000000	231506.000000
mean	1.559048	1.664332
std	1.200421	1.717384
min	1.000000	0.000000
25%	1.000000	1.100000
50%	1.000000	1.200000
75%	2.000000	1.500000
max	200.000000	60.000000

FIGURE 4 – Résumé statistique de la base de données

A traver le résumé statisque, on peut voir que la **quantité maximum vendue est 200**. Cette quantité nous paraît intriguante en ce sens, comme nous pouvons le voir ci-desssous l'article concernée par cette vente est : **CAFE OU EAU**, nous allons donc procéder à la **suppression de cet outlier**.

3.2 Feature Engineering

Afin de **mieux saisir les habitudes de consommation des clients de notre boulangerie**, il nous a paru pertinent de créer de nouvelles variables afin d'enrichir notre base de données.

Tout d'abord nous avons choisi d'ajouter une variable nommée **total amount spent** qui correspond au **montant total dépensé pour une transaction** en se basant sur les quantités et le prix unitaire des articles achetées.

Cette variable nous permettra de mieux appréhender les montants dépensées au sein de la boulangerie.

Ensuite, nous avons ajouté plusieurs variables liée à **la temporalité des ventes** en extrayant des variables initiales (dates et time) les informations suivantes :

- **year** correspondant à l'année de la vente
- **month** correspondant au mois de la vente
- **day of week** corrspondant au jour de la semaine où la vente a été réalisé
- **day time of purchase** correspondant à la plage horaire où la vente a été réalisée (ie [9,10) lorsque la commande a été passée à 9h38).Ces variables de temporalité nous permettront d'apprécier les comportements de vente des clients et de capter les différentes temporalités.

Enfin, il nous a paru pertinent d'ajouter une variable binaire nommée **holiday qui prend la valeur 1 lorsque la vente a été réalisée lors d'une période de vacances scolaires et 0 si celle-ci a été réalisée hors vacances scolaires**.

Pour créer cette variable, nous nous basons sur les périodes de vacances scolaires à Paris en 2021 et 2022 qui sont respectivement :

- Rentrée 2020 : mardi 1er septembre 2020.
- Vacances de la Toussaint : du samedi 17 octobre au dimanche 1er novembre 2020.
- Vacances de Noël : du samedi 19 décembre 2020 au dimanche 3 janvier 2021.
- Vacances d'hiver : du 13 février 2021 au 28 février 2021
- Vacances de printemps : du 17 avril au 2 mai pour la zone C.
- Vacances d'été : fin des cours le mardi 6 juillet 2021 au 2 septembre 2021
- Rentrée scolaire 2021 Jour de reprise : jeudi 2 septembre 2021
- Toussaint 2021 est du samedi 23 octobre au lundi 8 novembre 2021
- Noël 2021 est du samedi 18 décembre 2021 au lundi 3 janvier 2022
- Hiver 2022 est du du samedi 19 février au lundi 7 mars 2022
- Printemps 2022 (Pâques) est du samedi 23 avril au lundi 9 mai 2022
- Été 2022 (grandes vacances) est du du jeudi 7 juillet au jeudi 1er septembre 2022

	date	time	ticket_number	article	quantity	unit_price	total_amount_spent	day_of_week	year	month	holiday	day_time_of_purchase
0	2021-02-01	1900-01-01 08:38:00	150040.0	BAGUETTE	1.0	0.90	0.90	Lundi	2021	Fevrier	0	[07-08)
1	2021-02-01	1900-01-01 08:38:00	150040.0	PAIN AU CHOCOLAT	3.0	1.20	3.60	Lundi	2021	Fevrier	0	[07-08)
2	2021-02-01	1900-01-01 09:14:00	150041.0	PAIN AU CHOCOLAT	2.0	1.20	2.40	Lundi	2021	Fevrier	0	[08-09)
3	2021-02-01	1900-01-01 09:14:00	150041.0	PAIN	1.0	1.15	1.15	Lundi	2021	Fevrier	0	[08-09)
4	2021-02-01	1900-01-01 09:25:00	150042.0	TRADITIONAL BAGUETTE	5.0	1.20	6.00	Lundi	2021	Fevrier	0	[08-09)

FIGURE 5 – Affichage de la base après transformations

Création de la variable cible

Le but principal de notre projet est la **quantité journalière vendue** au sein de la boulangerie. Toutefois, ;cette variable n'existe pas en l'état dans notre base de données, il nous faut de ce fait la créer sur la base des variables 'quantity' et 'date' donnant les quantités unitaires vendues ne donnant pas la dimension totale que nous recherchons.

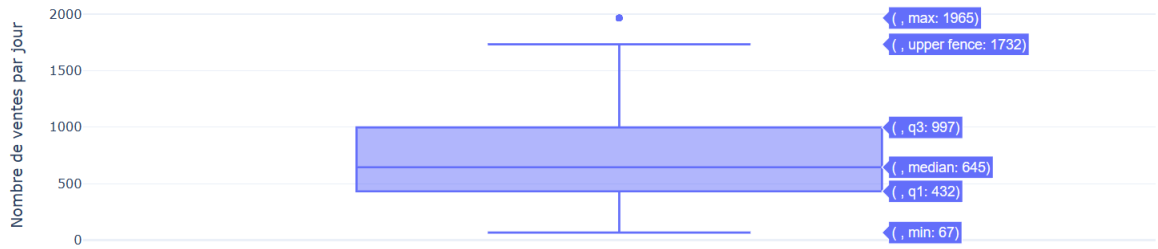

FIGURE 6 – Distribution de la variable cible

A traver le boxplot qui répresente la distribution de notre variable cible, on peut voir que la **quantité maximum vendue par jout est 1965 unités**. La mediane de la distribution se situe à 645 et le minimum se truve à 67 unités vendues par jour.

3.3 A la recherche des insights au sein de notre boulangerie

Afin de mieux comprendre notre jeu de données et anticiper les différentes tendances de consommation des clients de notre boulangerie, nous avons choisi de tourner notre analyse exploratoire des enjeux vers la connaissance des articles phares de la boulangerie mais également vers l'évolution des ventes de la boulangerie.

Ces informations nous permettront d'identifier les facteurs explicatifs de la quantité vendue de notre boulangerie, sujet de notre modèle prédictif qui sera mis en place dans la section suivante.

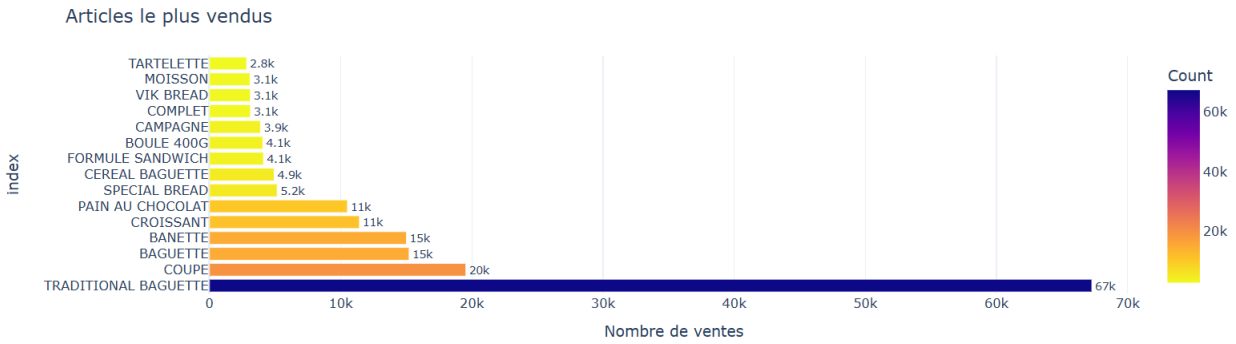

FIGURE 7 – Articles le plus vendus

Parmi les articles les plus vendus au sein de la boulangerie, **en emblème de la consommation française** on retrouve **la baguette tradition** qui **monopolise les ventes de la boulangerie**.

On retrouve dans le **top 5** : la baguette, la banette, la coupe et le croissant qui concurrence ensemble la primatie de notre baguette tradition nationale.

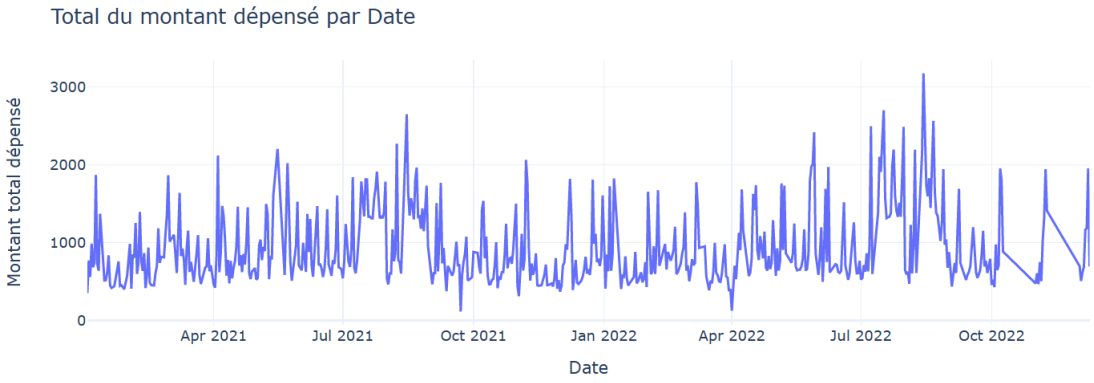

FIGURE 8 – Total du montant dépensé par date

A travers ce graphique de l'évolution des montants dépensées au sein de la boulangerie entre 2021 et 2022, on peut voir que **de plus gros montant ont été dépensé en 2022 en comparaison à 2021 même si on peut détecter une certaine tendance et saisonnalité dans les montants dépensés**.

On pourra ainsi décomposer notre série en tendance et saisonnalité.

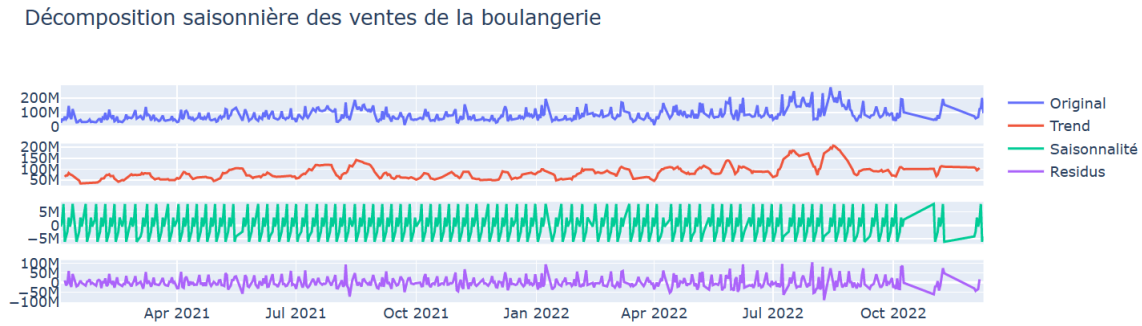

FIGURE 9 – Décomposition saisonnière

A travers ce graphique qui représente la décomposition saisonnière des ventes au sein de la boulangerie, on peut distinguer 3 grands phénomènes :

- Notre série comporte **une tendance claire haussière et baissière** selon la période de l'année considérée ce qui témoigne des **variations des ventes au sein de la boulangerie**.
- **La saisonnalité influe majoritairement à la hausse sur les ventes de la boulangerie**.
- La **composante résiduelle n'a pas un efet majeur sur les ventes de notre série** car elle est toujours en moyenne à 0 sauf quelques **pics en Septembre 2021 et Septembre 2022**.

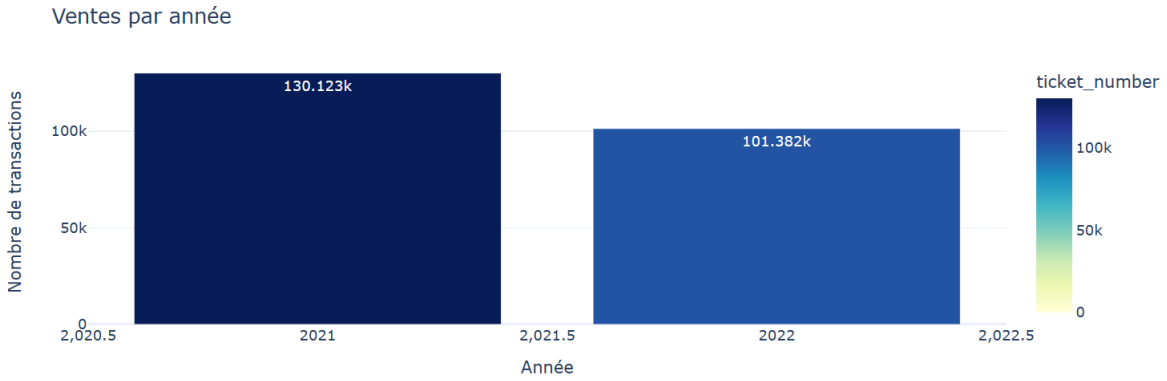

FIGURE 10 – Ventes par année

On peut voir à l'aide du graphique ci-dessus que **la boulangerie à réalisé le plus de ventes en 2021 avec 131 420 ventes** en comparaison avec **l'année 2022 où ce nombre de vente baisse de 22%**.

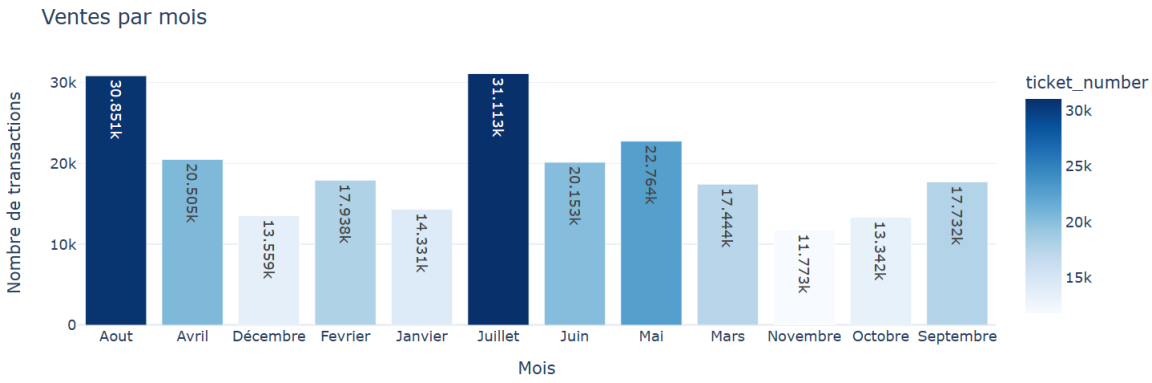

FIGURE 11 – Ventes par mois

A travers le graphique ci-dessus, on remarque que **la boulangerie vend ses plus grandes quantités durant la période estivale à savoir au mois de Juillet et Août** surêment car **c'est une période où la France accueille le plus de touristes** ainsi cette période permet à la boulangerie de vendre en plus grande quantité que le reste de l'année.

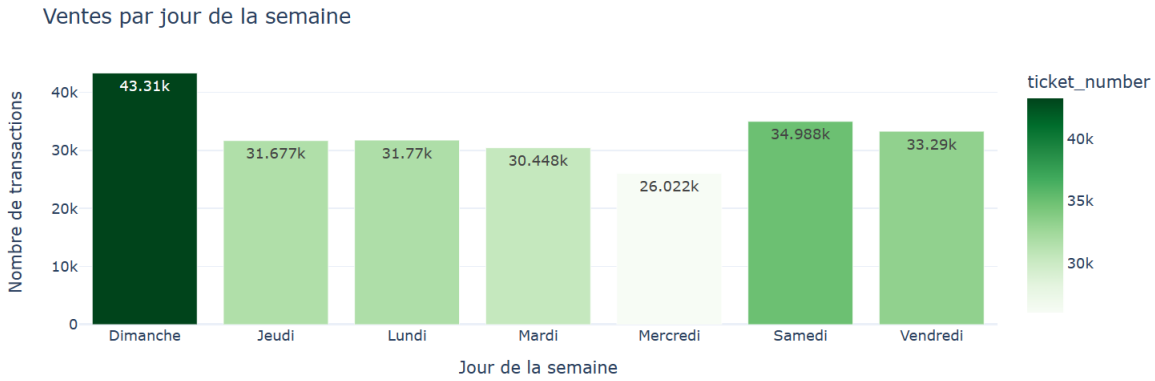

FIGURE 12 – Ventes par jour de la semaine

A travers ce graphique, nous pouvons constater que **le plus gros des ventes au sein de notre boulangerie sont réalisées du vendredi au dimanche avec le dimanche comme jour clé de vente**.

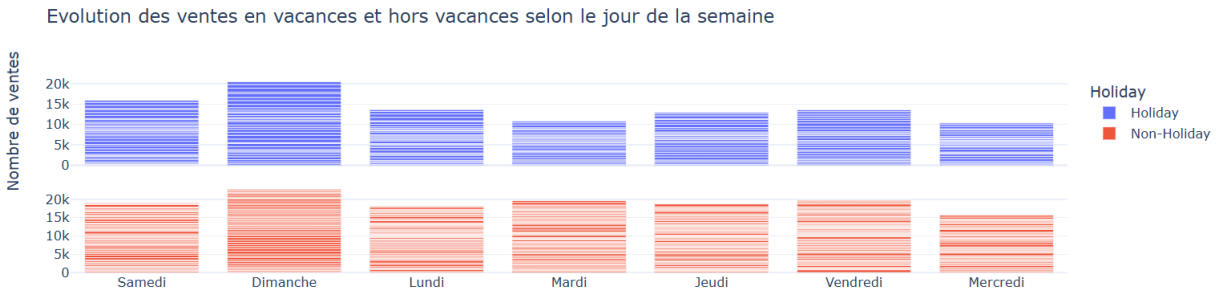

FIGURE 13 – Evolution des ventes en vacances

A travers ce graphique, nous pouvons détecter des tendances dans le comportement des clients selon que nous nous positionnons en périodes de vacances ou non :

— Les **ventes sont plus importantes lors des périodes de vacances** car on peut voir que plus de 15.000 ventes sont réalisées en moyenne.

— On constate que indistinctement de la période concernée, le **nombre maximal des ventes sont réalisées le dimanche**

— Hors vacances scolaires, **en fin de semaine le nombre de ventes réalisées augmentent** respectivement le jeudi et le vendredi comparées au reste du début de semaine.

4 Préparation des données pour la modélisation

Pour préparer des données pour être fournies à l'algorithme de ML/DL, nous avons choisit de garder que les variables numériques. Pour cela, nous avons introduit une nouvelle variable **daily total amount spent** qui répresente le montant total dépensé par jour.

Ensuite, nous avons agregé les données par jour pour avoir une seule ligne par jour.

date	daily_quantity_sold	holiday	daily_total_amount_spent
2021-01-02	43380.0	1	63612.00
2021-01-03	189312.0	1	268290.60
2021-01-04	97090.0	0	148667.40
2021-01-05	277123.0	0	408395.05
2021-01-06	121716.0	0	201301.80

FIGURE 14 – Affichage de la base de données pour la modélisation

Après l'aggrégation, nous nous retrouvons avec 600 lignes et 3 colonnes.

Concernant le split en train et test, nous avons choisit de diviser la table en 70% train et le 30% restant en test.

Nous avons préparer les données de façon distinct selon l'algorithme implémenté, vous retrouverez l'ensemble des préparations au sein du notebook du projet.

4.1 Analyse de stabilité de la variable cible

4.1.1 Test de stationnarité

La stationnarité d'une série temporelle est une hypothèse fondamentale dans de nombreux modèles statistiques.

Cette hypothèse signifie que les propriétés statistiques d'une série temporelle, telles que sa moyenne, sa variance et son autocorrélation, ne changent pas au fil du temps. Si une série temporelle n'est pas stationnaire, il peut être difficile de faire des prévisions précises car les propriétés statistiques de la série peuvent changer au fil du temps, ce qui conduit à des prévisions biaisées ou peu fiables.

Dans notre cas, nous avons choisit de faire 3 tests de stationnarité :

ADF test	**KPSS test**	**PP test**
statistique : -5.052	statistique : 0.215	statistique : -5.099
p-value : 1.746e-05	p-value : 0.1	p-value : 0.0001349
La série est stationnaire.	La série est non-stationnaire.	La série est non-stationnaire.

On choisit la règle de 3, donc nous pouvons conclure que notre target est stationnaire.

4.1.2 ACF et PACF

Afin de déterminer les ordres de variables passées à retenir pour expliquer la quantité journalière vendue au sein de la boulangerie, on se sert de l'autocorrélation partielle.

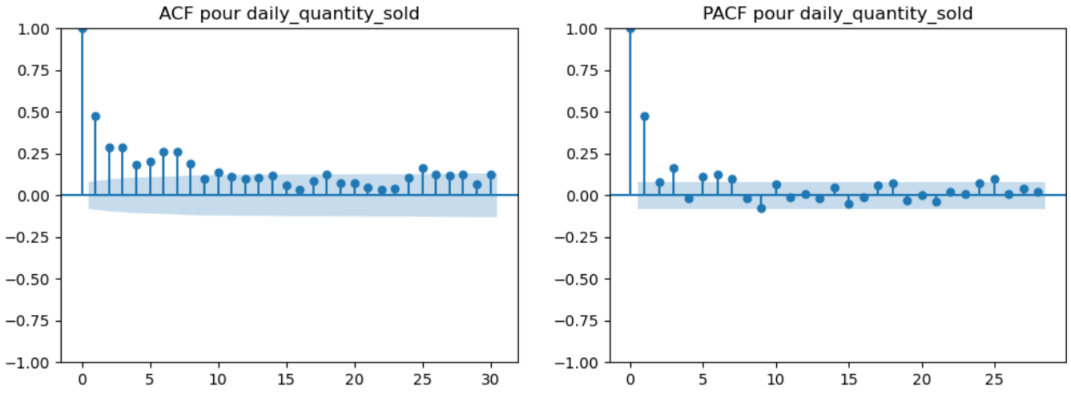

FIGURE 15 – ACF et PACF pour la variable cible

A travers notre autocorrélation partielle (PACF), on peut conclure que à partir de deux valeurs passées de la quantité journalière vendue (2 lags) on peut prédire la quantité journalière qui sera venue à la date t.

5 Sélection de différents modèles

5.1 Critères de sélection

Dans le cadre de la prévision des quantités vendues au sein de notre boulangerie, nous avons choisi d'implémenter trois algorithmes :

— Un modèle **SARIMAX** (Seasonal AutoRegressive Integrated Moving Average with Exogenous variables) : une extension du modèle ARIMA qui prédit les valeurs futures d'une série temporelle sur certains aspect de la structure statistique de la série à savoir sa partie **autorégressive** (AR) et de ses **moyennes glissantes** (MA). A l'inverse du modèle ARIMA, le **SARIMAX prend en compte la saisonnalité présente dans les séries** ce qui la rend d'autant plus intéréssante dans notre cadre d'étude de prévision des quantités vendues au sein de la boulangerie. De plus, ce modèle peut intégrer des variables explicatives supplémentaires (facteurs externes) susceptibles d'avoir un impact sur la série temporelle, comme par exemple les données météorologiques.

— Un **Gradient Boosting** est un bon choix pour la prévision de séries temporelles car il peut gérer des relations non linéaires, incorporer des caractéristiques externes, gérer des données manquantes, est robuste aux valeurs aberrantes et est une méthode d'ensemble qui combine les prédictions de plusieurs modèles faibles pour produire une prédiction forte.

— Un **Prophet** qui est un algorithme populaire de forecast de séries temporelles développé par Facebook. Il s'agit d'un **modèle de régression additive qui inclut les effets de tendance, de saisonnalité et de vacances, ainsi que des régresseurs supplémentaires, pour faire des prévisions**. Dans notre cadre, il est intéréssant de l'implémenter car il permet de faire des **prédictions à haute accuracy** tout en laissant une **bonne interprétabilité** du modèle à ses utilisateurs. De plus, son **horizon de prévision est très long et flexible** selon les besoins de l'utilisateur.

5.2 Choix de la métrique

Afin d'évaluer les performances de notre modèle, étant dans un cadre de prévision d'une quantité nous avons choisi comme métrique la **MAPE (mean absolute percentage error)** mesure la **différence en pourcentage entre les valeurs prévues et réelles, ce qui est une mesure pertinente pour évaluer les modèles de prévision dans des contextes commerciaux**.

6 Evaluation des performances

6.1 SARIMAX

Dans un premier temps, nous avons choisi de prédire les quantités journalières au sein de notre boulangerie à l'aide d'un modèle **SARIMAX**. C'est **un des modèles les plus utilisées en séries temporelles** car il permet de prendre en compte la composante saisonnière au sein des séries.

Avec ce modèle, nous obtenons une **MAPE de 0.848 indique qu'en moyenne, les prédictions du modèle sont d'environ 84% inférieures aux valeurs réelles**.

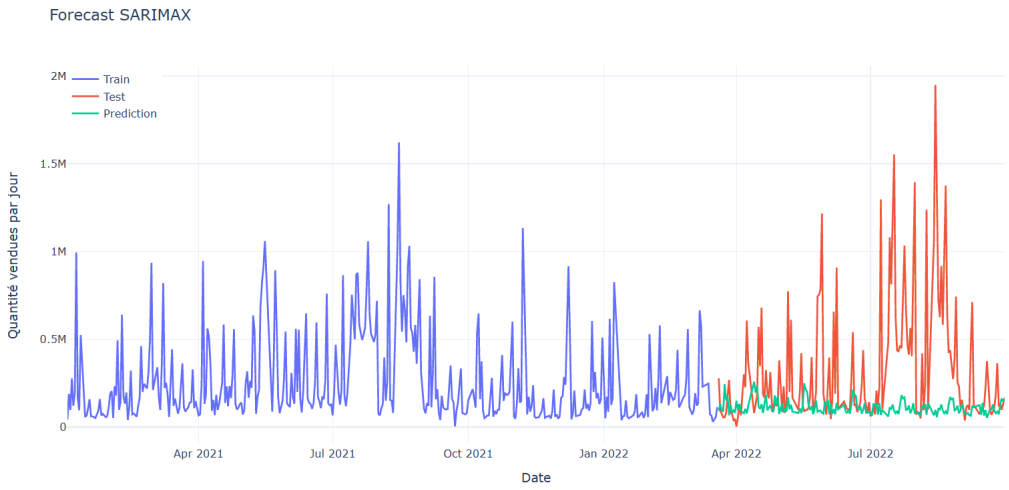

FIGURE 16 – Forecast SARIMAX

Comme affiché dans le graphique ci-dessus, on peut donc voir qu'ils n'arrivent pas à bien prédire les quantités vendues, nous pensons que cela est dû au fait que nous avons sélectionné les paramètres du modèle SARIMAX à l'aide de l'autocorrélation partielle du fait d'un manque de ressource (RAM) pour effectuer une sélection automatique des paramètres optimaux ce qui a sûrement conduit aux sous performances du modèle.

6.2 Gradient Boosting

Notre tâche de prédiction consiste à estimer les quantités de ventes quotidiennes, ce qui nécessite l'utilisation d'un modèle capable de prédire une variable cible continue. Nous avons opté pour le **Gradient Boosting Regressor**, une technique d'apprentissage automatique qui peut être utilisée pour résoudre des problèmes de régression en entraînant plusieurs modèles d'arbres de décision en série.

Avec ce modèle, nous obtenons un **MAPE de 0.1028 indique qu'en moyenne, les prédictions du modèle sont d'environ 10,28% inférieures aux valeurs réelles**.

Par exemple, si la quantité quotidienne réelle vendue est de 100, la prédiction du modèle serait, en moyenne, distante de 10,28% de la valeur réelle, qui est d'environ 10,28 unités.

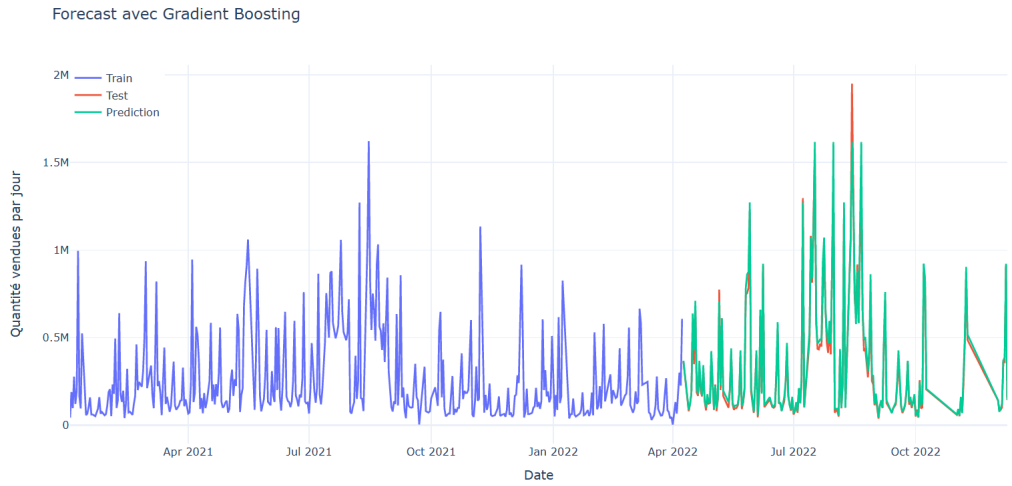

FIGURE 17 – Forecast Gradient Boosting

On peut voir à travers le graphique ci-dessus que **les prédictions de notre modèle de Gradient Boosting sont plutôt très précises** car elles collent aux valeurs réelles de l'échantillon de test.

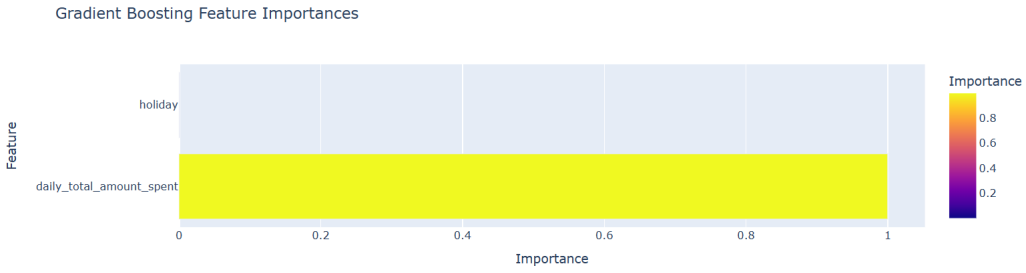

FIGURE 18 – Feature Importance Grandient Boosting

On peut voir que la variable qui contribue le plus à la prédiction des ventes journalières est : **'daily total amount spent'** correspondant au montant journalier dépensé par les clients au sein de la boulangerie. L'influence de la variable **'holiday'** est comme on peut le voir minime malgré l'idée qu'on pourrait avoir de son importance.

6.3 Prophet

Le denier modèle testé est un modèle de Deep Learning qui s'apppelle Prophet. C'est **un des modèles les plus utilisées en séries temporelles** car il permet de prendre en compte la composante saisonnière au sein des séries. Nous avons à l'aide de celui-ci tenter de prédire les quantités vendues notamment pour les 5 articles les plus vendus au sein de la boulangerie comme par exemple la baguette.

Avec ce modèle, nous obtenons une **MAPE de 0.368 (en moyenne) indique qu'en moyenne, les prédictions du modèle sont d'environ 37% inférieures aux valeurs réelles**.

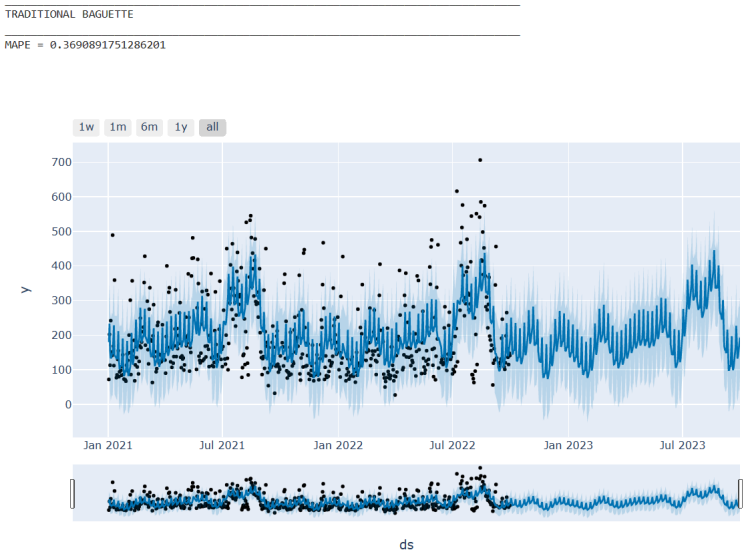

FIGURE 19 – Example Forecast Prophet

Comme affiché dans le graphique ci-dessus, on peut donc voir que le modèle n'arrive pas à bien prédire les quantités vendues, nous pensons que cela est dû au fait qu'il y a des valeurs aberrantes que le modèle n'arrive pas à capter.

7 Discussion autour des performances

TABLE 1 – Evaluation des performances des modèles

	MAPE
SARIMAX	0,848
Gradient Boosting	0,0903
Prophet	
- Traditional Baguette	0.369
- Croissant	0.465
- Pain au chocolat	0.309
- Banette	0.317
- Baguette	0.387

Tout d'abord, concernant le modèle **SARIMAX** on peut voir selon lee tableau ci-dessus que c'est le modèle qui performe le moins bien. Ses mauvaises performances peuvent provenir de différentes raisons :

- Une première raison est **la mauvaise spécification des ordres et des hyperparamètres dans le modèle**.
- Il nous semble que **le modèle n'est pas pas en mesure de capturer la saisonnalité des ventes journalières de notre boulangerie** ainsi que les tendances inhérentes ce qui cause un bruit dans les prédicitions.
- Egalement, une autre raison peut être dû aux valuers aberrantes ou aux **événements inattendus**, comme nous pouvons voir à cause de certains pics à certaines dates (par exemple juillet et aout), donc le modèle n'arrive pas à capter ces situations et donc il peut produire des prévisions inexactes.

Toutefois, nous pouvons constater que pour un horizon court de temps (1 mois), les prévions du SARIMAX ne sont pas mauvaises.En effet, nous pouvons constater que pour le mois de mars 2022, la MAPE pour le modèle SARIMAX s'eleve à 0,079. Cela signifie qu'en moyenne, les prédictions du modèle sont d'environ 7,9% inférieures aux valeurs réelles.

8 Synthèse, conclusion et pistes d'améliorations

En somme, ce projet de modélisation des quantités vendues au sein de la boulangerie nous a permis de manipuler des séries temporelles et d'appréhender les concepts inhérents à la tendance et la saisonnalité propres aux séries temporelles.

Il nous a également permis d'évaluer les performances de modèles divers telles que de machine learning avec le Gradient Boosting, de deep learning avec le Prophet et enfin plus statistiques comme le SARIMAX.

Plusieurs constats peuvent être tirer à l'issue de l'évaluation des performances respectives des modèles implémentés :

— **Les performances des modèles aurait pu être meilleur si à l'aide d'un historique plus riche nous avions effectué des prédictions mensuelles voire hebdomadaire** car cela donne une meilleure idée de la tendance des ventes au sein de la boulangerie. Ces performances mensuelles améliorées ont notamment été visibles sur les différents modèles tels que SARIMAX et Gradient Boosting où notre métrique est meilleure mensuellement.

— Le modèle de Gradient Boosting performe globalement mieux que les autres modèles selon nous car il n'inclue pas comme les autres modèles les notions de tendance ou de saisonnalité pouvant bruiter les prédictions.

— Les prédicitions par type d'article comme nous l'avons fait dans le Prophet ne donne pas de bonnes prédictions mais celles-ci sont encourageantes quant à la précision que peut avoir le modèle Prophet sur un grand historique de données.

Enfin, comme pistes d'améliorations nous aurions :

— Dans un premier temps pu optimiser nos modèles à l'aide de la recherche des hyperparamètres optimaux à l'aide d'algorithme comme le **gridsearchCV** pour le Gradient Boosting ou le **RandomizedSearchCV** pour le Prophet.

— Dans un second temps, **nous aurions pu inclure dans notre modèle des informations sur la température journalière** qui influe sur les articles vendues en boulangerie.

— Egalement, nous aurions pu inclure en plus des périodes de vacances, les jours fériés afin d'enrichir notre connaissance des comportements des clients de la boulangerie.