**Note méthodologique : preuve de concept**

## **Dataset retenu**

**Sources et préparation initiale :**

Pour la préparation initiale des [données](https://www.kaggle.com/datasets/iamprateek/wallmart-sales-forecast-datasets), je me suis appuyé sur le travail réalisé dans le notebook Kaggle suivant : [Walmart Sales Forecasting](https://www.kaggle.com/code/aslanahmedov/walmart-sales-forecasting) .

Cette analyse a permis d'étudier les principales étapes d'ingénierie des données, notamment :

* + L'importation et la consolidation des différents fichiers sources.
  + La création de nouvelles fonctionnalités comme des variables temporelles (e.g., jour de la semaine, mois, vacances).
  + Une première analyse exploratoire pour identifier les patterns clés dans les séries temporelles.

Après avoir étudié et optimisé le dataset initial, le dataset final est constitué des colonnes suivantes :

* + `*Store*` : Identifiant unique du magasin.
  + `*Date*` : Date de la semaine correspondante.
  + `*Weekly\_Sales*` : Chiffre d'affaires hebdomadaire.

***Les données ont été pré traitées pour inclure :***

* + Conversion des dates en un format temporel standardisé.
  + Échantillonnage hebdomadaire pour assurer une fréquence uniforme.
  + Gestion des valeurs manquantes via le report des valeurs précédentes (***padding***).

**Statistiques du dataset final**

* ***Période couverte*** : Deux ans.
* ***Nombre de magasins*** : 45.
* ***Fréquence***: Hebdomadaire.
* ***Taille finale*** : Une série temporelle par magasin, associée à un identifiant unique (`item\_id`), permettant des prévisions spécifiques à chaque magasin.

## **Les concepts de l’algorithme récent**

Chronos est un cadre innovant qui adapte les modèles de langage, tels que T5, à la prévision de séries temporelles en transformant les valeurs numériques en une séquence de tokens discrets.

[arXiv](https://arxiv.org/pdf/2403.07815v3?utm_source=chatgpt.com)

**1. Prétraitement des données :**

* **Mise à l'échelle** : Chaque série temporelle est normalisée pour réduire les variations de magnitude entre les séries. Chronos utilise la mise à l'échelle par la moyenne, où chaque valeur est divisée par la moyenne des valeurs absolues de la série historique.
* **Quantification** : Les valeurs mises à l'échelle sont ensuite discrétisées en les assignant à des bins (intervalles) prédéfinis, chaque bin correspondant à un token unique. Cette transformation permet de représenter les données continues en une séquence de tokens discrets.

**2. Entraînement du modèle :**

* **Architecture** : Chronos s'appuie sur des architectures de modèles de langage existantes, comme T5, sans modifications majeures. Le modèle est entraîné à prédire le token suivant dans la séquence, en minimisant la perte d'entropie croisée entre les tokens prévus et les tokens réels.
* **Données d'entraînement** : Le modèle est préentraîné sur un vaste ensemble de données publiques de séries temporelles, enrichi par des données synthétiques générées via des processus gaussiens. Cette approche améliore la généralisation du modèle à des séries temporelles variées.

**3. Prévision :**

* **Inférence** : Lors de la prévision, le modèle génère des tokens de manière autoregressive, c'est-à-dire en prédisant chaque token successif basé sur les tokens précédemment générés. Ces tokens sont ensuite convertis en valeurs numériques réelles en inversant le processus de quantification et de mise à l'échelle.
* **Distribution probabiliste** : En échantillonnant plusieurs trajectoires de tokens, Chronos est capable de produire une distribution probabiliste des prévisions, offrant ainsi une estimation de l'incertitude associée aux prédictions.

En résumé, Chronos adapte efficacement les modèles de langage à la prévision de séries temporelles en transformant les données continues en séquences de tokens, permettant ainsi l'utilisation directe des architectures de modèles de langage pour des tâches de prévision.

**La modélisation**

### **1. Méthodologie de Modélisation**

Chronos s’appuie sur les principes fondamentaux des modèles de langage (Language Models) tout en les adaptant aux spécificités des séries temporelles. Voici les étapes clés :

#### **1.1 Prétraitement des Données**

1. **Mise à l’échelle des séries temporelles** : Chaque série est normalisée pour réduire les écarts de magnitude. Chronos applique une mise à l’échelle par la moyenne, où chaque valeur est divisée par la moyenne des valeurs absolues de la série.
2. **Quantification** : Les données numériques sont discrétisées en bins prédéfinis. Chaque intervalle correspond à un token unique, permettant de transformer les données en une séquence de tokens exploitables par les modèles de langage.

#### **1.2 Architecture du Modèle**

1. **Modèle de base** : Chronos repose sur T5 (Text-to-Text Transfer Transformer), une architecture conçue pour les tâches de séquences textuelles. Aucune modification majeure n'est apportée à l’architecture.
2. **Entraînement autoregressif** : Le modèle est entraîné à prédire chaque token de manière séquentielle en minimisant une fonction de perte d’entropie croisée.
3. **Robustesse aux variations** : Pour améliorer la généralisation, le modèle est pré-entraîné sur un mélange de séries temporelles publiques et de données synthétiques générées via des processus gaussiens.

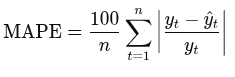
#### **1.3 Inférence**

1. **Prévision token par token** : Lors de l’inférence, le modèle génère des tokens de manière autoregressive, puis les convertit en valeurs numériques réelles via un processus d’inversion (quantification inverse et dénormalisation).
2. **Distribution probabiliste** : Chronos produit plusieurs trajectoires prédites afin de caractériser l’incertitude des prévisions.

### **2. Métrique d'Évaluation**

Pour évaluer les performances, la métrique **Mean Absolute Percentage Error (MAPE)** est utilisée. Elle est couramment employée en prévision pour mesurer l’erreur relative moyenne entre les valeurs réelles et prédites.

* **Formule** :



où yt​ est la valeur réelle, y^​t​ est la valeur prédite, et n est le nombre total de points.

* **Interprétation** :
  + Une MAPE faible indique une meilleure précision du modèle.
  + C’est une métrique intuitive qui exprime l’erreur en pourcentage, facilitant la compréhension et la comparaison entre modèles.
* **Limitation** : La MAPE est sensible aux petites valeurs de yt​. Pour les séries temporelles comportant des valeurs proches de zéro, des transformations log-scaled peuvent être utilisées.

### **3. Démarche d’Optimisation**

Chronos adopte une approche méthodique pour optimiser les performances du modèle :

#### **3.1 Entraînement et Validation**

1. **Validation croisée** : Les séries temporelles sont divisées en fenêtres temporelles afin de garantir une séparation stricte entre les données d’entraînement et de test.
2. **Optimisation des hyperparamètres** :
   * Les intervalles de discrétisation (quantification) et la taille des bins sont ajustés pour équilibrer la granularité et la précision.
   * Les hyperparamètres du modèle T5 (taille des couches, taux d’apprentissage, etc.) sont optimisés via des méthodes comme l’optimisation bayésienne.

#### **3.2 Gestion de l’incertitude**

1. **Échantillonnage multiple** : Chronos génère plusieurs séquences pour estimer une distribution de probabilités.
2. **Ajustement des prévisions** : Les trajectoires sont pondérées pour minimiser le risque d’erreurs extrêmes.

4.

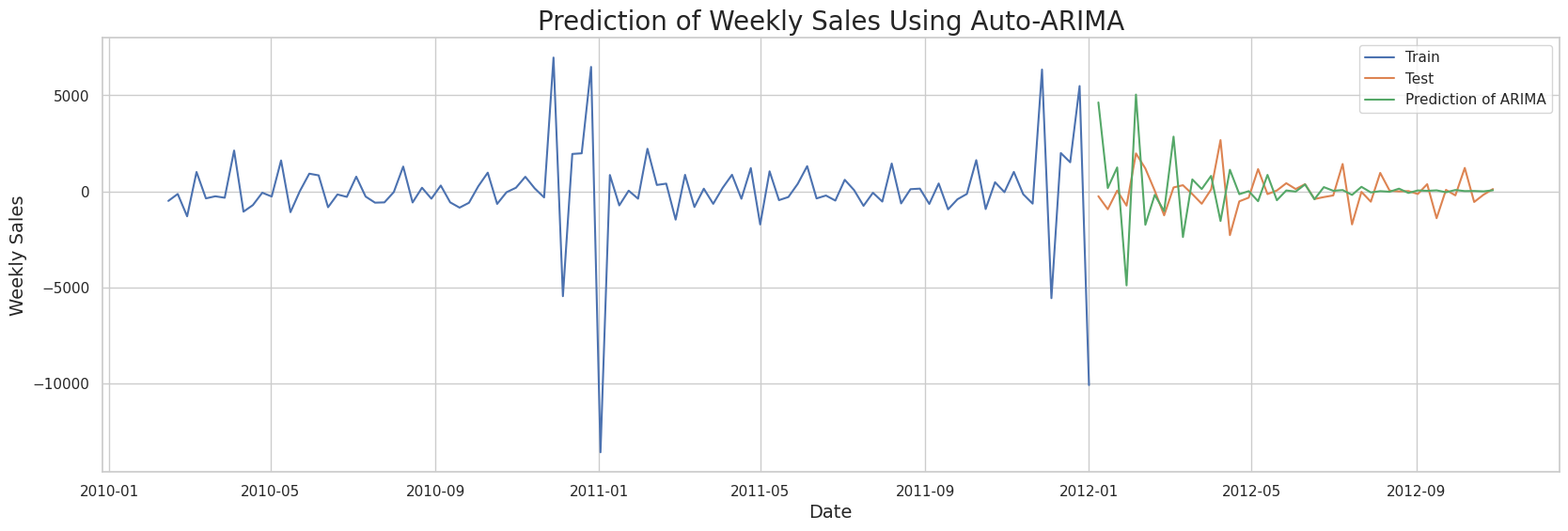
**Une synthèse des résultats : AutoARIMA et CHRONOS**

L’objectif de cette synthèse est de comparer les performances des deux techniques sur la base du MAPE (Mean Absolute Percentage Error) et des graphiques de prédiction.

#### 1. Résultats Comparés

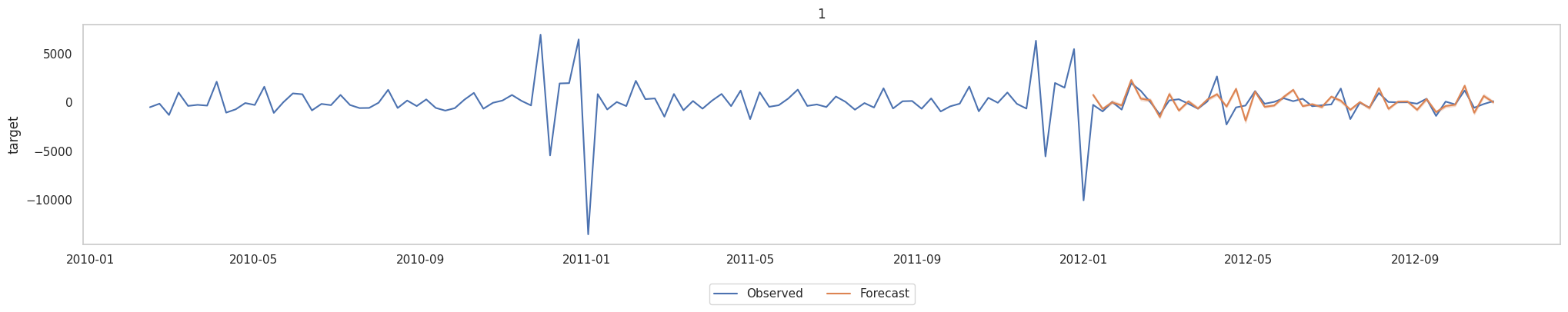
##### **AutoARIMA**

1. **Méthodologie** :
   * Modèle ajusté sur les données historiques après différentiation saisonnière pour stabiliser la variance.
   * La sélection automatique des paramètres (p, d, q) optimise la précision.
2. **Performance** :
   * **MAPE** calculé : 6397.56%.
   * AutoARIMA présente des performances décevantes comme on peut le constater sur le graphique ci-dessous.



##### **Chronos**

1. **Méthodologie** :
   * Conversion des données continues en séquences discrètes (étiquettes) via la quantification.
   * Modèle basé sur ***chronos\_tiny***, ajusté pour prédire chaque token de manière autoregressive.
   * Prévisions finales obtenues par inversion des transformations initiales.
2. **Performance** :
   * **MAPE** calculé : ***35***% (testé sur les mêmes données que AutoARIMA).
   * Chronos offre une flexibilité accrue pour des séries multivariées et permet de gérer des dynamiques temporelles complexes.



#### 2. Analyse

1. **Précision** :
   * Chronos présente un MAPE inférieur à celui d’AutoARIMA, indiquant une meilleure précision des prévisions.
   * Les prédictions d’AutoARIMA tendent à être biaisées dans des cas de variations rapides, où Chronos s’avère plus robuste.
2. **Applicabilité** :
   * AutoARIMA est idéal pour des séries univariées avec des motifs saisonniers clairs.
   * Chronos excelle dans des scénarios où les données présentent des relations non linéaires ou des caractéristiques complexes.

#### 4. Conclusion

Les résultats montrent que Chronos surpasse AutoARIMA en termes de précision pour des séries temporelles complexes. Cependant, son adoption implique des coûts computationnels élevés et une courbe d’apprentissage plus importante.

En conclusion :

* **AutoARIMA** reste une option viable pour des problèmes simples, où la simplicité est prioritaire.
* **Chronos** constitue une approche d’avant-garde pour relever les défis des prévisions modernes, surtout dans des environnements riches en données multivariées.

**L’analyse de la feature importance globale et locale du nouveau modèle**

### 1. **Contexte de la Feature Importance**

Dans les modèles traditionnels de machine learning (comme les arbres de décision ou les modèles linéaires), la feature importance est une mesure permettant d'évaluer l'influence relative de chaque variable d'entrée sur les prédictions. Cela est particulièrement utile dans des contextes multivariés où chaque variable joue un rôle spécifique.

Dans le cadre des séries temporelles, l’importance des variables peut être envisagée pour :

* Identifier les contributeurs majeurs aux prédictions (importance globale),
* Comprendre les effets locaux pour une période spécifique (importance locale).

### 2. **Limites des Transformers Pré-entraînés dans les Séries Temporelles**

Les modèles Transformers pré-entraînés, comme Chronos, introduisent des limitations spécifiques en matière d’analyse de l’importance des variables, pour plusieurs raisons fondamentales :

#### a) **Absence de variables explicatives dans les séries univariées**

Dans le cadre de votre projet, les données sont univariées : chaque série temporelle est définie uniquement par l’historique des ventes d’un magasin. Il n’existe pas de variables explicatives additionnelles (comme la météo ou des indicateurs économiques) pouvant être évaluées pour leur importance.

#### b) **Représentation implicite des patterns temporels**

Les Transformers tokenisent les valeurs des séries temporelles en séquences discrètes et utilisent des embeddings pour représenter les données. Ces représentations apprennent des relations complexes entre les observations passées et futures, mais elles ne permettent pas de quantifier directement l'importance des variables, car :

* L'information est dispersée à travers des dimensions latentes,
* L’attention multi-têtes se concentre sur des séquences spécifiques sans expliciter l'impact des variables d'entrée.

#### c) **Absence de mécanismes intégrés pour expliquer les décisions**

Contrairement aux modèles comme les arbres de décision ou les réseaux neuronaux LIME/SHAP-ready, les Transformers ne possèdent pas d’interprétabilité native :

* L’attention peut être utilisée pour observer les poids entre les tokens, mais cela ne représente pas directement l'importance des variables.
* Les variables étant tokenisées et agrégées, leur rôle individuel est dilué.

### 3. **Pourquoi l’importance des variables n’est pas applicable ici**

Dans ce projet, le modèle Chronos ne permet pas d’évaluer l’importance des variables pour les raisons suivantes :

1. **Modèle univarié :** Il n'y a qu'une seule variable (Weekly\_Sales), et donc aucune comparaison possible entre plusieurs contributeurs.
2. **Approche pré-entraînée :** Les modèles pré-entraînés apprennent des relations générales sur de larges ensembles de données, et les embeddings générés ne reflètent pas une importance individuelle des tokens.
3. **Focus sur les relations temporelles :** L’objectif principal du modèle est de capturer les patterns temporels (tendances, saisonnalités, résidus) et non d’évaluer l’influence de variables additionnelles.

### 4. **Approches Alternatives**

Bien que le modèle Chronos ne permette pas une analyse directe de l’importance des variables, il est possible d’envisager des alternatives pour évaluer ou compenser ce manque :

* **Ajouter des variables explicatives :** Introduire des variables multivariées comme les conditions météo, les jours fériés ou les promotions permettrait de construire des modèles où l’importance des variables pourrait être évaluée.
* **Utiliser des modèles explicables en parallèle :** En complément de Chronos, des modèles comme XGBoost ou des réseaux neuronaux interprétables peuvent être utilisés pour évaluer l’importance des variables.
* **Interpréter les distributions prédictives :** Analyser les distributions probabilistes générées par Chronos (plutôt que l’importance des variables) pour comprendre les incertitudes dans les prévisions.

### Conclusion

Dans le cadre de votre projet, le modèle Chronos ne permet pas d’évaluer l’importance globale ou locale des variables, car :

1. Le modèle est univarié et ne dispose que d’une seule entrée (Weekly\_Sales).
2. Les Transformers pré-entraînés agrègent les données sous forme de séquences tokenisées, rendant difficile l’interprétation des variables individuelles.
3. L’objectif principal est la capture des patterns temporels complexes, et non l’explicabilité des variables.

Pour améliorer l’interprétabilité, des approches multivariées ou hybrides pourraient être envisagées dans des projets futurs.

**Les limites et les améliorations possibles**

*Présentez les limites et les améliorations envisageables pour gagner en performance et en interprétabilité de l'approche de modélisation, en 1 page maximum.*