

# projet deep learning

réalisé par: Ayer Med Amine 2023-2024

---

Rapp ort

- 1. Introduction générale
- 2. Cadre général du projet
  - 2.1 Introduction
  - 2.2 État de l'art
    - 2.2.1 Étude de l'existant
    - 2.2.2 Revue de quelques applications similaires
    - 2.2.3 Critique de l'existant
    - 2.2.4 Solution envisagée
  - 2.3 Méthodologie de travail
    - 2.3.1 Gestion du projet avec CRISP-DM
  - 2.4 Conclusion
  -
- 3. Processus décisionnel
  - 3.1 Introduction
  - 3.2 Définition
  - 3.3 Objectifs
  - 3.4 Étapes du projet
  - 3.5 Outils
  - 3.6 Schéma dimensionnel
  -
- 4. Deep Learning
  - 4.1 Introduction
  - 4.2 Bibliothèques utilisées
  - 4.3 Prétraitement des données
  - 4.4 Modélisation
  - 4.5 Prédiction
  - 4.6 Résultats et analyse
  - 4.7 Conclusion
  -
- 5. Déploiement
  - 5.1 Introduction
  - 5.2 Préparation de l'environnement de production
  - 5.3 Développement de l'application Django
  - 5.4 Avantages de l'utilisation de Django
  - 5.5 Conclusion
  -

## Liste des Figures

1. Cycle de vie de CRISP-DM
2. Site de récupération des données
3. Nettoyage des données sur Jupyter
4. Analyse préliminaire avec des Tests Statistiques
5. Visualisation sur PowerBi
6. Prédiction par le modèle de Machine Learning
7. Logo Talend
8. Schéma conceptuel du Data Warehouse
9. Exemple de prétraitement
10. Exemple de prétraitement
11. Exemple de prétraitement
12. Entraînement du modèle 1
13. Entraînement du modèle 2
14. Entraînement du modèle 3
15. Entraînement du modèle 4
16. Code de la prédiction
17. Évaluation des modèles

## Liste des tables

1	Applications similaires. . . . .	6
---	----------------------------------	---

# 1 Introduction générale

Le domaine de la technologie de l'informatique, en particulier l'informatique décisionnelle et le deep learning, joue un rôle crucial dans la numérisation de la gestion d'un volume important de données, offrant une solution optimale pour faciliter le travail et le rendre plus précis en exploitant les points d'amélioration de l'organisation du travail.

Dans le secteur du commerce de détail, les experts cherchent constamment à maximiser leur efficacité en numérisant les systèmes de prévision des ventes et des prix. L'informatique décisionnelle, combinée à des techniques avancées de deep learning, occupe une place prépondérante dans ce processus. Les étapes clés comprennent la compréhension de la logique métier, l'analyse de l'historique des ventes, le choix du meilleur modèle de Deep Learning pour la prévision, et enfin, la phase de test visant à obtenir les meilleurs résultats.

Le deep learning, en exploitant des réseaux de neurones profonds, permet une analyse plus approfondie des données, identifiant des modèles complexes et des relations non linéaires. Dans le contexte des prévisions de ventes et de prix, cette approche peut améliorer la précision des résultats, offrant ainsi des informations plus pertinentes pour les décideurs du secteur de la vente au détail.

La prédiction des tendances de vente et des fluctuations de prix, grâce à l'intégration du deep learning, devient encore plus puissante, permettant une adaptation plus agile aux changements du marché. Les entreprises du secteur du commerce de détail recherchent activement des moyens d'incorporer ces avancées technologiques pour rester compétitives et réactives.

Notre projet s'inscrit dans cette perspective, visant à développer une application de prévision des tendances de vente et des prix dans le secteur du commerce de détail, en utilisant des techniques avancées de deep learning en plus de l'historique des données des 30 dernières années. L'objectif principal de ce projet est d'identifier le meilleur modèle de prédiction en minimisant les erreurs, contribuant ainsi à une meilleure planification des stocks et à des décisions de tarification plus éclairées grâce à l'utilisation de l'informatique décisionnelle et du deep learning.

Ce rapport détaille les différentes étapes de notre travail, mettant en avant l'importance cruciale du deep learning dans le contexte de la gestion des données et des prévisions dans le secteur du commerce de détail.

## 2 Cadre général du projet

### 2.1 Introduction

La réalisation d'une application requiert inévitablement une étude préliminaire approfondie. Cette phase essentielle englobe l'analyse, l'évaluation et la critique des applications similaires, posant ainsi les fondations nécessaires à la proposition ultérieure d'une solution pertinente. Dans cette optique, il est suggéré de débiter en exposant le fonctionnement actuel du système, tout en examinant quelques applications semblables et en évaluant leurs critiques. Cette approche permettra d'obtenir une vision d'ensemble cruciale pour le développement d'une solution plus avancée et adaptée.

### 2.2 Etat de l'art

#### 2.2.1 Etude de l'existant

Il existe des solutions de prédiction des tendances dans le secteur du commerce de détail. Parmi Description : RetailTrend est une application de prédiction des tendances dans le commerce de détail. Elle utilise des modèles avancés pour anticiper les évolutions des ventes, des préférences des consommateurs et des variations de prix, aidant les entreprises à optimiser leur stratégie de tarification et de stock pour rester compétitives.

#### 2.2.2 Revue de quelques applications similaires

La prévision des prix dans le secteur du commerce de détail représente un défi complexe. Cela implique la compréhension des dynamiques commerciales, la collecte de données historiques, et le développement de modèles de prévision basés sur l'apprentissage automatique. Malgré l'existence de solutions sur le marché, les modèles traditionnels présentent des limites. L'exploration de nouvelles approches, notamment l'intégration de l'intelligence artificielle, est essentielle pour améliorer la précision des prévisions. Tester ces modèles reste crucial pour identifier les meilleures configurations et minimiser les erreurs, assurant ainsi une adaptation continue aux évolutions du marché.

#### 2.2.3 Critique de l'existant

Notre application de prévision des tendances de vente et des prix dans le secteur du commerce de détail se distingue par son approche axée sur la qualité plutôt que la quantité. Plutôt que de proposer une multitude de fonctionnalités, nous nous concentrons sur l'essentiel, en fournissant un ensemble complet d'outils soigneusement conçus pour répondre aux besoins spécifiques des utilisateurs du commerce de détail. Notre objectif est de simplifier le processus de prévision des prix en offrant des fonctionnalités essentielles, telles que l'analyse approfondie de l'historique des données et l'utilisation de modèles avancés, pour permettre des décisions de tarification éclairées. Ainsi, notre application vise à répondre de manière exhaustive aux exigences des individus et des entreprises, contribuant à l'efficacité globale de leurs stratégies commerciales.

#### 2.2.4 Solution envisagée

Nous envisageons la mise en place d'un système de prédiction et de visualisation graphique des tendances de vente et des prix dans le secteur du commerce de détail. La gestion de ces fonctionnalités à travers notre application garantira les points suivants :

1. Prévission Améliorée : Notre application vise à perfectionner le processus de prévision, maximisant ainsi la précision et éliminant toute source potentielle d'erreur.
2. Visibilité Totale des Tendances : Permettant le suivi et la sauvegarde des tendances de vente et des fluctuations de prix dans l'historique des données, offrant une vue complète et chronologique.
3. Réponse Adaptée aux Besoins du Commerce de Détail : Gestion efficace des demandes et mise à jour en temps réel lors de la confirmation et de la mise en place des activités, assurant une réponse agile aux attentes des individus et entreprises du secteur.

#### 2.3 Méthodologie de travail

Le choix d'une méthodologie agile est largement répandu pour divers types de projets. La méthode agile repose sur un cycle de développement centré sur l'interaction avec le client. L'implication continue du client tout au long du projet offre une vision plus claire et améliore le travail au fur et à mesure.

Nous avons opté pour l'utilisation de la méthodologie CRISP-DM en raison de sa pertinence par rapport à nos exigences en matière d'informatique décisionnelle (BI). Cette méthodologie, qui se caractérise par ses étapes claires et itératives, s'aligne parfaitement sur nos besoins spécifiques, favorisant ainsi une approche systématique et efficace dans le développement de notre solution.

##### 2.3.1 Gestion du projet avec CRISP-DM

Travailler avec la méthodologie CRISP-DM implique la progression à travers six étapes clés dans la réalisation du projet :

1. Compréhension du Problème Métier : Cette étape vise à comprendre le contexte commercial et les objectifs du projet.
2. Compréhension des Données : Impliquant la collecte et l'exploration des données pour évaluer leur qualité et leur pertinence pour le projet.
3. Préparation des Données : Consistant à nettoyer et transformer les données en vue de les préparer pour l'analyse.
4. Modélisation : Création des modèles d'analyse de données répondant aux objectifs du projet.
5. Évaluation : Cette étape consiste à évaluer la qualité des modèles créés pour déterminer leur adéquation aux objectifs du projet.
6. Déploiement : Mettre en œuvre les modèles dans un environnement opérationnel.
7. Surveillance : Surveiller le système en production pour garantir que les résultats sont conformes aux attentes.

La planification du projet sert à organiser efficacement les activités pour atteindre les objectifs fixés. Cela commence par l'affectation des rôles CRISP-DM, suivi de la planification des réunions, de l'élaboration du backlog du produit, et enfin, du plan des livrables et des sprints.

La figure suivante illustre le cycle de vie du CRISP-DM :

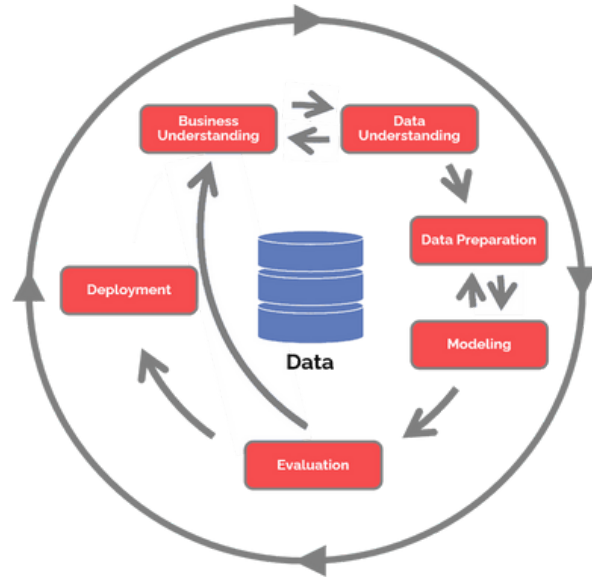


Figure 1: Cycle de vie de CRISP-DM

## 2.4 Conclusion

Dans ce chapitre, nous avons présenté le cadre général et le contexte du sujet. Nous avons également exposé une critique de l'existant, mettant en lumière les applications similaires, dont l'étude nous a guidés vers une solution adéquate répondant aux besoins du projet.



## 3 Processus décisionnel

### 3.1 Introduction

Pour assurer un pilotage efficace et faire face à la concurrence, l'adoption d'un système d'information décisionnel est primordiale. Cela permet aux décideurs de prendre des décisions éclairées. Dans le contexte de ces évolutions, il est essentiel de définir clairement ce qu'est l'informatique décisionnelle.

### 3.2 Définition

L'informatique décisionnelle, également connue sous le nom de Business Intelligence (BI), englobe un ensemble de techniques et d'outils permettant de collecter, d'analyser et de présenter des données, facilitant ainsi la prise de décisions éclairées au sein des entreprises. Son objectif principal est de fournir des informations précieuses et des perspectives afin d'aider les décideurs à comprendre les tendances, les opportunités et les défis auxquels leur entreprise est confrontée.

Les systèmes d'informatique décisionnelle intègrent diverses technologies telles que la modélisation des données, l'analyse de données, l'exploration de données, la visualisation de données, les tableaux de bord et les rapports. Ces systèmes sont utilisés pour suivre les performances de l'entreprise, identifier les tendances et les opportunités, et prendre des décisions basées sur des données concrètes plutôt que sur des conjectures ou des suppositions.

En résumé, l'informatique décisionnelle constitue un domaine clé de l'informatique visant à aider les entreprises à prendre des décisions plus informées et plus efficaces en utilisant des données et des analyses approfondies.

### 3.3 Objectifs

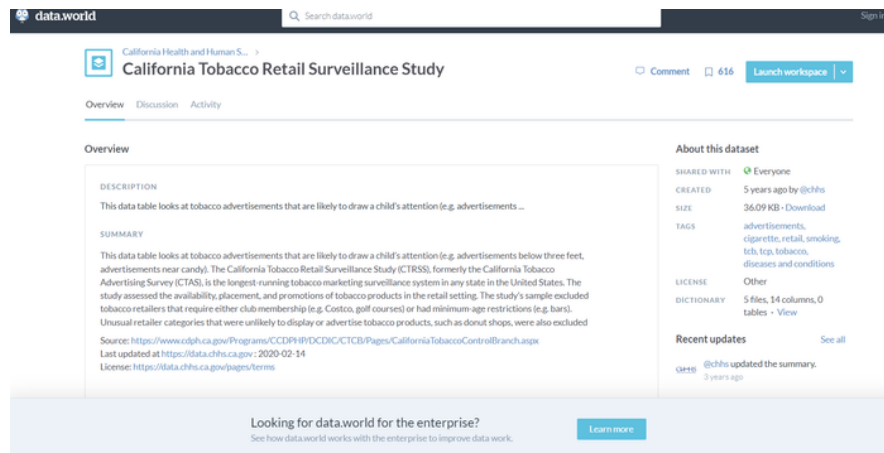
Les objectifs de l'informatique décisionnelle sont multiples, tous convergent vers l'objectif global de fournir des informations utiles pour aider les entreprises à prendre des décisions éclairées. Voici quelques-uns des objectifs les plus courants de l'informatique décisionnelle :

- **Collecter et Intégrer des Données** : L'informatique décisionnelle vise à collecter et intégrer des données provenant de différentes sources au sein de l'entreprise, telles que les données de ventes, de production, de marketing, et financières. Ces données sont souvent stockées dans des entrepôts de données ou des datamarts pour faciliter l'accès et la manipulation.
- **Analyser les Données pour Identifier des Tendances et des Opportunités** : L'utilisation de techniques d'analyse de données permet d'identifier des tendances et des opportunités cachées dans les données. Ces analyses incluent des méthodes telles que l'analyse de corrélation, l'analyse de régression, l'analyse de clustering, et d'autres encore.
- **Fournir des Informations en Temps Réel** : L'informatique décisionnelle a la capacité de fournir des informations en temps réel aux décideurs de l'entreprise, notamment à travers des tableaux de bord présentant les données clés et les indicateurs de performance.
- **Améliorer la Prise de Décision** : L'un des principaux objectifs de l'informatique décisionnelle est d'améliorer la prise de décision en fournissant des informations factuelles et des analyses approfondies aux décideurs de l'entreprise. Cela peut contribuer à des décisions plus claires, plus stratégiques, et, par conséquent, à une meilleure performance et une croissance plus rapide de l'entreprise.

### 3.4 Etapes du projet

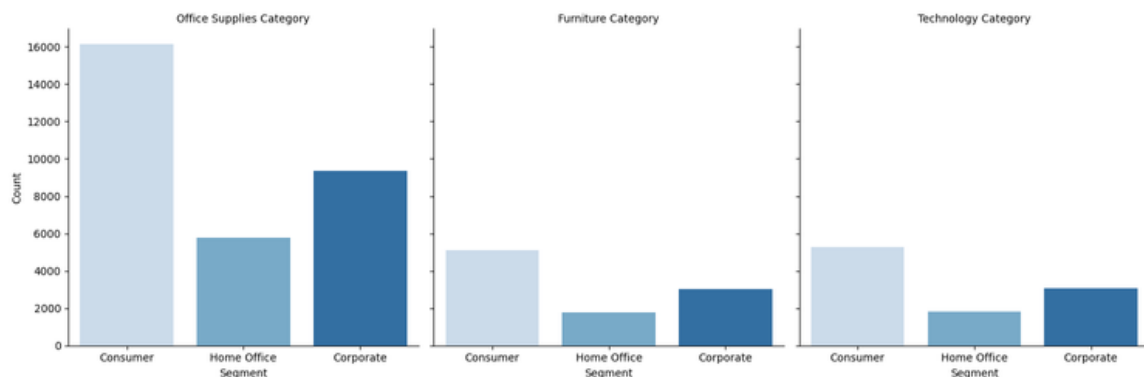
L'informatique décisionnelle implique plusieurs étapes pour permettre aux entreprises de collecter, analyser et présenter des données de manière à faciliter la prise de décisions éclairées. Voici les étapes typiques de l'informatique décisionnelle :

- Collecte de Données : La première étape de l'informatique décisionnelle consiste à collecter des données provenant de différentes sources, telles que des bases de données, des feuilles de calcul, des fichiers texte et des sources externes. Les données sont souvent centralisées dans un entrepôt de données pour faciliter l'accès et la gestion.



Site de Récupération des Données"  
(Figure 2)

Analyse de Données : La troisième étape de l'informatique décisionnelle consiste à analyser les données pour en tirer des informations utiles. Cela peut inclure l'utilisation d'outils de visualisation de données, d'outils de modélisation statistique, d'algorithmes d'apprentissage automatique, etc. L'objectif est d'explorer les données de manière approfondie, d'identifier des tendances, des modèles ou des informations significatives qui peuvent guider les décisions éclairées. Dans le contexte de la visualisation des données, cette étape joue un rôle clé pour transformer les données brutes en insights visuels compréhensibles.



```

import pandas as pd
from statsmodels.tsa.stattools import adfuller, kpss

# Assuming you have already loaded your DataFrame and set 'Order Date' as the index
# If not, convert 'Order Date' to datetime and set it as the index (as shown previously)

# Extract sales data for 2012 and 2014
sales_2012 = sliced_df_2012_2013['mean']

# Define a significance level (alpha)
alpha = 0.05

# Perform the ADF test for seasonality on 2012 data
adf_result_2012 = adfuller(sales_2012)
adf_p_value_2012 = adf_result_2012[1]

# Perform the KPSS test for seasonality on 2012 data
kpss_result_2012 = kpss(sales_2012, regression='c')
kpss_p_value_2012 = kpss_result_2012[1]

# Interpret and compare the test results
def interpret_test_result(p_value, test_name, year):
    print(f'{test_name} p-value for {year} data: {p_value:.4f}')
    if p_value < alpha:
        print(f'Reject the null hypothesis (H0): {year} data may have seasonality.')
    else:

```

"Analyse Préliminaire avec des Tests Statistiques"  
(Figure 4).

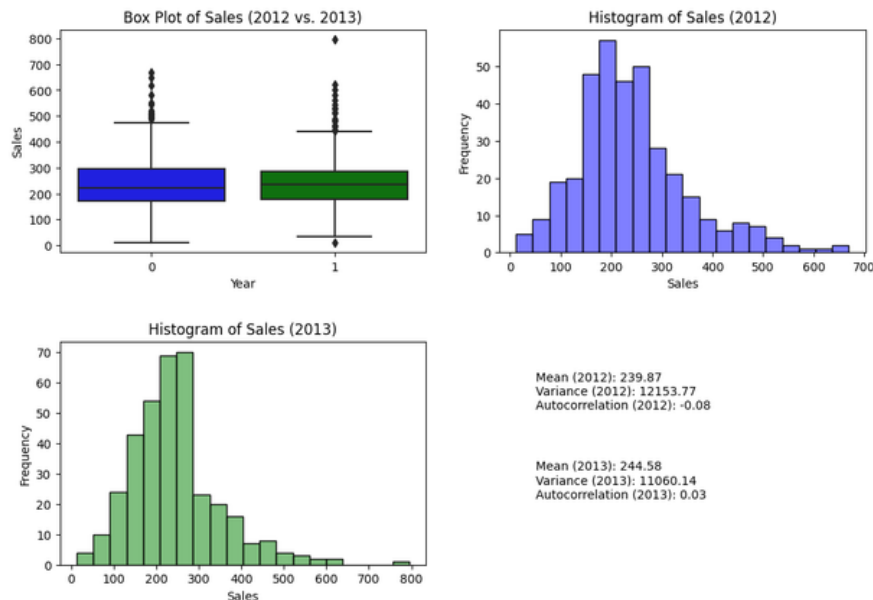
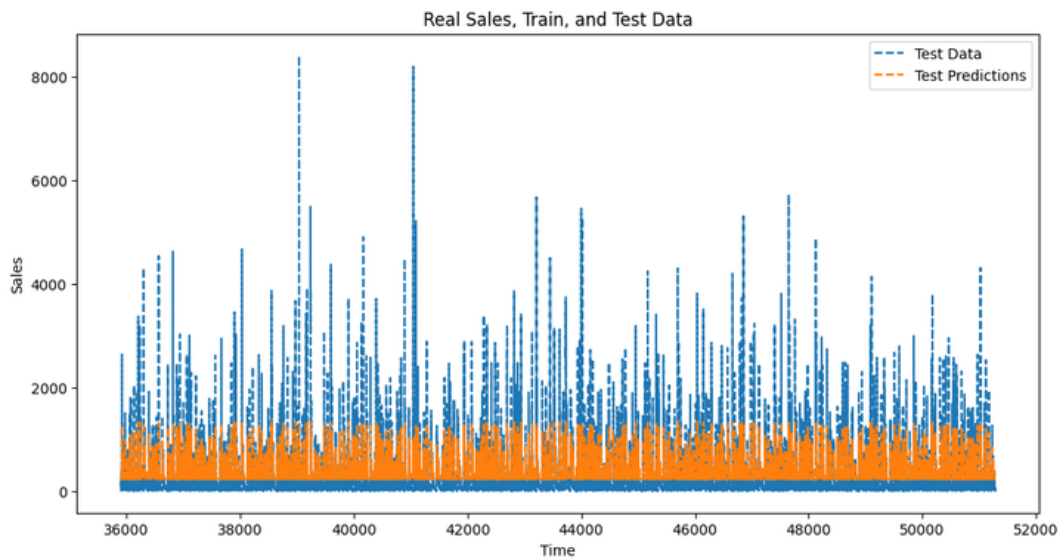


Figure 5: Etude de serie temporelle

Prise de Décision : La dernière étape de l'informatique décisionnelle consiste à utiliser les informations collectées et analysées pour prendre des décisions éclairées. Cette étape peut inclure la définition d'objectifs clairs, la création de plans d'action, la mise en œuvre de stratégies, etc. L'objectif final est d'utiliser les insights obtenus pour orienter les actions de manière efficace, influençant ainsi positivement les résultats et la performance de l'entreprise.



Prédiction par le Modèle de Deep Learning (Figure 6)

### 3.5 Outils

Google Colab est une plateforme de notebook basée sur le cloud qui permet d'écrire et d'exécuter du code Python de manière interactive. Elle offre un environnement de développement collaboratif, hébergé par Google, qui est largement utilisé dans le domaine de la science des données et de l'apprentissage automatique. Voici quelques points importants concernant l'utilisation de Google Colab dans ce contexte :

**Environnement Basé sur le Cloud :** Google Colab fournit un environnement de développement Python basé sur le cloud. Les utilisateurs peuvent exécuter des notebooks directement dans leur navigateur sans avoir besoin d'installer des outils locaux.

**Accès aux GPU :** Un avantage significatif de Google Colab est la possibilité d'accéder gratuitement à des ressources GPU. Cela est particulièrement utile pour les tâches gourmandes en calcul, telles que l'entraînement de modèles d'apprentissage automatique.

**Intégration avec Google Drive :** Colab est intégré à Google Drive, ce qui facilite le stockage et le partage de notebooks. Les utilisateurs peuvent également sauvegarder leurs travaux directement sur Google Drive.

**Bibliothèques et Frameworks :** Colab prend en charge de nombreuses bibliothèques et frameworks populaires utilisés en science des données, tels que NumPy, Pandas, TensorFlow, PyTorch, etc. Les utilisateurs peuvent installer des bibliothèques supplémentaires au besoin.

**Collaboration en Temps Réel :** Comme Google Colab est basé sur le cloud, plusieurs utilisateurs peuvent collaborer en temps réel sur un même notebook. Cela facilite le travail d'équipe et la discussion autour du code.

**Importation et Exportation de Données :** Colab permet l'importation et l'exportation faciles de données à partir de Google Drive, ce qui simplifie le processus de gestion des ensembles de données.

**Exécution de Code par Lots :** Les notebooks peuvent être exécutés par lots, permettant ainsi une exécution automatisée de sections spécifiques du code à des intervalles définis.

**Visualisation de Données :** Colab prend en charge la visualisation interactive de données à l'aide de bibliothèques telles que Matplotlib et Seaborn.



Figure 7: Logo Google colab

## 4 Deep Learning

### 4.1 Introduction

**L'apprentissage profond, une branche de l'intelligence artificielle, utilise des réseaux neuronaux profonds pour automatiser des tâches complexes en apprenant à partir de données. Les réseaux profonds, avec leurs multiples couches, apprennent des caractéristiques hiérarchiques, permettant des applications étendues comme la vision par ordinateur et la reconnaissance vocale. Ce modèle nécessite souvent de grandes quantités de données pour un entraînement efficace et tire parti de l'accélération GPU. L'apprentissage profond a révolutionné plusieurs domaines de l'IA, ouvrant la voie à des avancées significatives.**

### 4.2 Bibliothèques utilisées

Python se classe parmi les langages de programmation les plus prisés pour les projets de machine learning. Il est reconnu pour sa simplicité, sa polyvalence et la vaste communauté de développeurs qui ont élaboré de nombreuses bibliothèques afin de simplifier les tâches courantes en science des données et en apprentissage automatique.

Ce rapport présentera quelques-unes des bibliothèques Python les plus fréquemment utilisées dans notre projet de machine learning. Ces bibliothèques ont été sélectionnées en fonction de leur popularité, de leur efficacité et de leur facilité d'utilisation



- NumPy : une bibliothèque pour le calcul scientifique qui fournit des structures de données de base pour le traitement des tableaux multidimensionnels.



- opérations de traitement de données complexes, etc.



- Matplotlib : une bibliothèque pour la création de graphiques et de visualisations de données en 2D.



- Seaborn : une bibliothèque pour la visualisation de données statistiques, offrant des fonctionnalités pour la création de graphiques en 2D plus avancés.



- Scikit-learn : une bibliothèque pour l'apprentissage automatique, proposant une gamme d'algorithmes pour la classification, la régression, le clustering, etc.



- SciPy : SciPy est une bibliothèque Python utilisée pour le calcul scientifique et technique.

#### 4.3 Prétraitement des données

Le prétraitement des données, également appelé prétraitement des données en anglais, revêt une importance cruciale au sein du processus d'apprentissage automatique. Il englobe les étapes de nettoyage, de transformation et de préparation des données brutes, les rendant ainsi exploitables par les algorithmes d'apprentissage automatique.

L'objectif du prétraitement des données est d'améliorer la qualité des données en entrée pour les algorithmes d'apprentissage automatique. Cela se fait en éliminant les problèmes potentiels tels que les biais, les valeurs extrêmes, ou les données manquantes, susceptibles d'affecter négativement les performances des modèles de prédiction. En résumé, le prétraitement des données constitue une étape clé de l'apprentissage automatique, permettant d'optimiser les performances des modèles en améliorant la qualité et la pertinence des données en entrée.

```
dfs=df.drop(['Order Date', 'Ship Date', 'Customer Name', 'Product Name', 'State', 'City', 'Country', 'Market', 'Region', 'Row ID', 'Order ID', 'Customer ID'])

dfs.columns

categorical_columns = ['Segment', 'Category', 'Sub-Category', 'Order-Priority']
dfs = pd.get_dummies(dfs, columns=categorical_columns, drop_first=True)
dfs.columns

X = dfs.drop('Sales',axis=1)
X.columns
```

Figure 9: exemple de prétraitement

```
# Extract the 'Sales' values
sales = dfs[['Sales']]

# Normalize the 'Sales' data
scaler = MinMaxScaler()
sales_scaled = scaler.fit_transform(sales)

# Reshape the data for LSTM
sales_data = np.array(sales_scaled).reshape(-1, 1)

# Split the data into training and testing sets
split = int(0.7 * len(sales_data))
X_train = sales_data[:split]
X_test = sales_data[split:]

# Prepare the target variable (Y) by shifting the data by one time step
Y_train = X_train[1:]
Y_test = X_test[1:]

# Reshape the data for LSTM
X_train_resaped = np.reshape(X_train[:-1], (X_train.shape[0] - 1, 1, 1))
X_test_resaped = np.reshape(X_test[:-1], (X_test.shape[0] - 1, 1, 1))
```

Figure 10: exemple de prétraitement

## 4.4 Modélisation

La modélisation occupe une position centrale au sein du processus d'apprentissage automatique. Elle implique le choix et l'entraînement d'un modèle statistique à partir des données prétraitées pour accomplir une tâche spécifique, telle que la classification, la régression ou la prédiction.

Le choix du modèle dépend de la nature de la tâche à accomplir et des caractéristiques des données en entrée. Il existe une diversité de modèles d'apprentissage automatique, allant des modèles simples comme la régression linéaire, aux modèles plus complexes tels que les réseaux de neurones.

```
# Load your time series data into a DataFrame
# Replace 'your_data.csv' with your actual data file
# Assuming your DataFrame has a DateTimeIndex and a 'Sales' column

# Step 1: Split the data into training and test sets
train_size = int(len(sliced_df_2012_2013) * 0.8) # 80% for training
train, test = sliced_df_2012_2013 .iloc[:train_size], sliced_df_2012_2013 .iloc[train_size:]

# Step 2: Use auto_arima to find the best ARIMA parameters
auto_model = auto_arima(train['mean'], seasonal=False, trace=True)

# Get the best ARIMA order (p, d, q)
best_order = auto_model.get_params()['order']

# Step 3: Fit the final ARIMA model with the best parameters
model = ARIMA(train['mean'], order=best_order)
model_fit = model.fit()

# Step 4: Make forecasts
forecast_values = model_fit.forecast(steps=len(test))

# Step 5: Evaluate the model's performance
mse = mean_squared_error(test['mean'], forecast_values)
print(f"Mean Squared Error (MSE): {mse:.2f}")

# Step 6: Visualize the forecasts
plt.figure(figsize=(12, 6))
plt.plot(train.index, train['mean'], label='Train Data')
plt.plot(test.index, test['mean'], label='Test Data')
plt.plot(test.index, forecast_values, label='ARIMA Forecast', color='red')
plt.title('Sales Forecast using ARIMA')
plt.xlabel('Date')
```

Figure 12: entraînement du model 1



```

WARNING:tensorflow:Layer lstm_6 will not use cuDNN kernels since it doesn't meet the criteria. It will use a generic GPU kernel.
Model: "sequential_6"

```

Layer (type)	Output Shape	Param #
lstm_6 (LSTM)	(None, 50)	10400
dense_6 (Dense)	(None, 1)	51

```

Total params: 10451 (40.82 KB)
Trainable params: 10451 (40.82 KB)
Non-trainable params: 0 (0.00 Byte)

```

Figure 13: architecture du model 2

## 4.5 Prédiction

La prédiction constitue l'une des applications principales de l'apprentissage automatique. Elle implique l'utilisation d'un modèle entraîné sur des données pour anticiper une sortie ou une valeur future à partir de données d'entrée.

Une fois que le modèle a été entraîné et évalué, il peut être employé pour effectuer des prédictions sur de nouvelles données. Les données d'entrée sont fournies au modèle, qui utilise les relations apprises pendant l'entraînement pour anticiper la sortie souhaitée.

```
y_pred=model.predict(x_test)
```

Figure 16: Code de la prédiction

## 4.6 Résultats et analyse

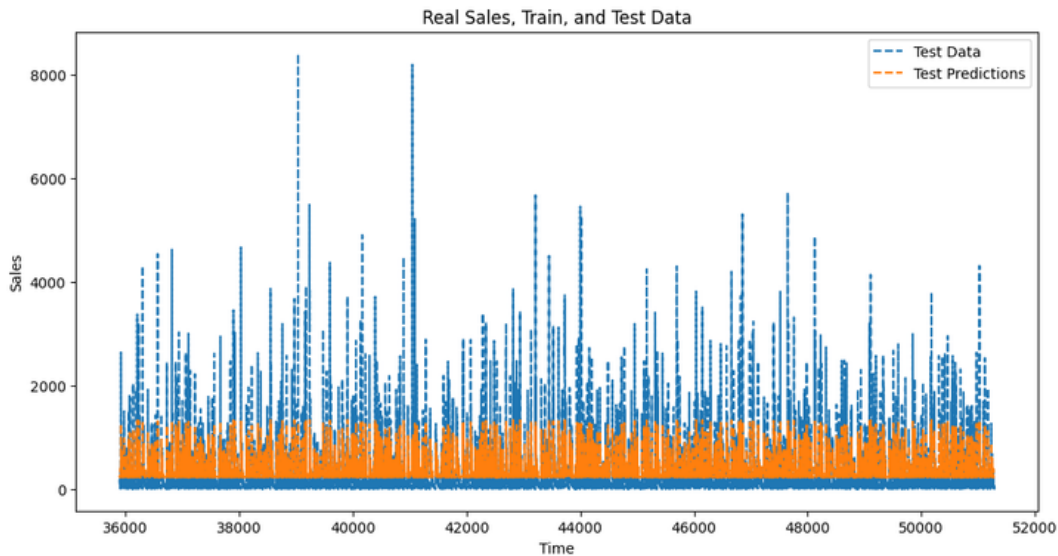


Figure 17: Evaluation des modèles

## 4.7 Conclusion

Dans cette partie on a présenté la partie Deep Learning de notre projet, qui servira come "input" pour la partie déploiement.

# 5 Déploiement

## 5.1 Introduction

La partie Déploiement est une étape importante dans la mise en place d'un système d'Intelligence Artificielle (IA) et de Business Intelligence (BI). Elle concerne la mise en production des modèles de Machine Learning et des tableaux de bord de BI pour permettre aux utilisateurs finaux d'interagir avec les données et de prendre des décisions informées.

## 5.2 Préparation de l'environnement de production

La première étape du déploiement implique la préparation de l'environnement de production. Cette phase englobe l'installation et la configuration des bibliothèques Python, la création de la base de données, et la configuration des paramètres de l'application Flask.

Framework :



Flask: Flask est un framework web gratuit et open-source, basé sur Python, qui suit le modèle architectural modèle-vues. Il est maintenu par la Flask Software Foundation.

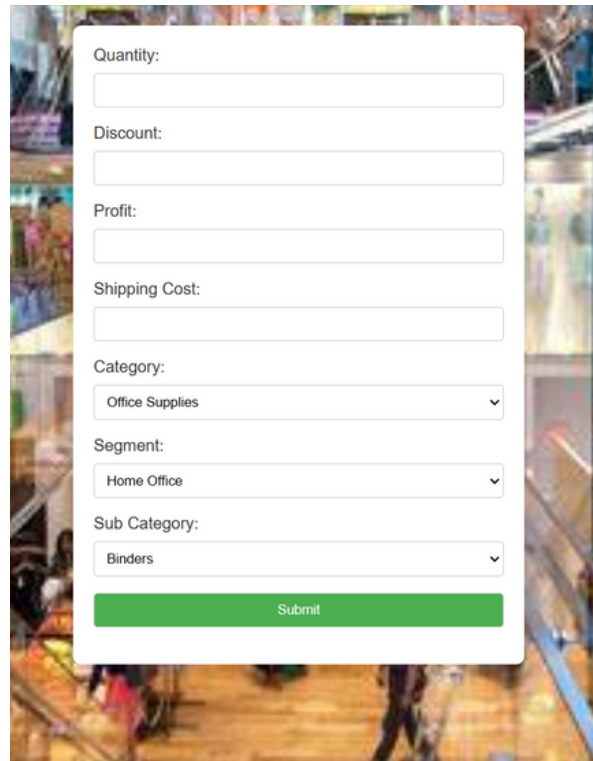
## 5.3 Développement de l'application Flask

La deuxième étape du déploiement implique le développement de l'application Flask en production, intégrant les modèles de Deep Learning ainsi que les tableaux de bord de Business Intelligence. Cette phase comprend la création des vues.

pour les tableaux de bord, la création des modèles pour la base de données et la configuration des URLs pour l'application.

## 5.4 Avantages de l'utilisation de Flask

L'utilisation de Django pour le déploiement des modèles de Machine Learning et des tableaux de bord de BI présente de nombreux avantages, notamment une réduction du temps de développement, une simplicité d'utilisation et une flexibilité pour la création d'applications Web.



Quantity:

Discount:

Profit:

Shipping Cost:

Category:  
Office Supplies ▼

Segment:  
Home Office ▼

Sub Category:  
Binders ▼

Submit

exemple de deployment

## 5.5 Conclusion

En conclusion, la section sur le déploiement de ce rapport a démontré que la mise en production des modèles de Deep Learning représente une étape cruciale dans l'établissement d'un système d'IA . Le déploiement requiert une préparation minutieuse de l'environnement de production, ainsi qu'une gestion et une surveillance régulières afin d'assurer un fonctionnement optimal et une maintenance efficace.

## 6 Conclusion Générale

La section sur le Deep Learning de ce rapport ainsi que celle sur la prediction de prix ont mis en évidence l'importance de l'analyse de données pour soutenir les entreprises dans la prise de décisions éclairées. Les techniques de Deep Learning ont été appliquées à un ensemble de données liées au prix de produit afin de prédire les variables cibles telles que le prix actuel et également pour identifier les facteurs les plus significatifs qui influent sur ces variables.