Pour un projet de **prévision des ventes avec visualisation interactive**, il est essentiel de choisir un jeu de données riche, réaliste, avec une composante temporelle et des variables explicatives (prix, promotions, événements, etc.). Voici une proposition concrète avec des détails :

**🎯 Jeu de données recommandé :**

**"Rossmann Store Sales"**

📁 Disponible sur : Kaggle - Rossmann Store Sales

**🧾 Description du jeu de données :**

Rossmann est une chaîne de pharmacies allemande. Ce jeu de données contient les **ventes quotidiennes de plus de 1 000 magasins** sur plusieurs mois.

**Contenu principal :**

* Store : Identifiant du magasin
* Date : Date (permet la modélisation temporelle)
* Sales : Ventes (cible à prédire)
* Customers : Nombre de clients (utile pour comprendre les fluctuations)
* Open : Si le magasin était ouvert (0/1)
* Promo : S’il y avait une promotion (0/1)
* StateHoliday : Jour férié d’État
* SchoolHoliday : S’il y avait des vacances scolaires
* StoreType et Assortment : Type de magasin et assortiment

**✅ Pourquoi ce dataset est pertinent ?**

* **Riche en variables explicatives** (promotions, jours fériés, type de magasin)
* **Temporel** → idéal pour les séries chronologiques (Prophet, LSTM, ARIMA…)
* **Multisite** → possibilité de comparaison et segmentation par magasin
* **Taille suffisante** pour entraîner des modèles ML/DL
* **Données réalistes** issues d’une entreprise réelle

**📊 Idées de visualisations interactives :**

* Évolution des ventes dans le temps (par magasin, région, etc.)
* Impact des promotions ou vacances sur les ventes
* Comparaison des performances entre différents types de magasins
* Dashboard interactif avec **Plotly Dash**, **Streamlit** ou **Power BI**

**📦 Bonus : Suggestions d’outils et modèles**

* **Prévision** :
  + Prophet (par Facebook)
  + ARIMA/SARIMA
  + LSTM (Deep Learning)
  + XGBoost/LightGBM avec ingénierie temporelle
* **Visualisation interactive** :
  + Python : Plotly, Dash, Streamlit, Altair
  + Web : Power BI, Tableau, Google Data Studio

Souhaites-tu que je t’aide à démarrer le projet (ex. : notebook de base, visualisation Streamlit, ou analyse exploratoire) ?

**Vous avez dit :**

puis je faire les previsions avec python

**ChatGPT a dit :**

Oui, absolument ! ✅ Tu peux **faire des prévisions de ventes avec Python** de manière très puissante. Voici un aperçu clair de **comment** procéder, avec des **outils adaptés**, une **méthodologie** et des **exemples de bibliothèques**.

**🧠 Étapes pour faire des prévisions de ventes avec Python :**

**1. Chargement et préparation des données**

* Bibliothèques : pandas, numpy
* Objectifs :
  + Convertir les dates (Date) au format datetime
  + Trier les données chronologiquement
  + Gérer les valeurs manquantes
  + Ajouter des variables dérivées (jour de la semaine, mois, etc.)

python

CopierModifier

import pandas as pd

df = pd.read\_csv("train.csv")

df['Date'] = pd.to\_datetime(df['Date'])

df = df.sort\_values(by='Date')

**2. Analyse exploratoire (EDA)**

* Bibliothèques : matplotlib, seaborn, plotly
* Objectifs :
  + Visualiser les ventes dans le temps
  + Voir l'effet des promotions, jours fériés, etc.

python

CopierModifier

import matplotlib.pyplot as plt

store\_sales = df[df['Store'] == 1]

plt.plot(store\_sales['Date'], store\_sales['Sales'])

plt.title('Ventes du magasin 1')

plt.xlabel('Date')

plt.ylabel('Ventes')

plt.show()

**3. Prétraitement & ingénierie des features**

* Créer de nouvelles colonnes :
  + day\_of\_week, is\_weekend, month, trend, lag\_features, etc.
* Encodage des variables catégorielles

**4. Choix du modèle de prévision**

Tu as plusieurs options selon ton niveau et ton besoin :

**🔹 Option 1 : Prophet (facile et puissant)**

python

CopierModifier

from prophet import Prophet

df\_prophet = store\_sales[['Date', 'Sales']].rename(columns={'Date': 'ds', 'Sales': 'y'})

model = Prophet()

model.fit(df\_prophet)

future = model.make\_future\_dataframe(periods=30) # prévisions 30 jours

forecast = model.predict(future)

model.plot(forecast)

**🔹 Option 2 : ARIMA/SARIMA (statistiques classiques)**

python

CopierModifier

from statsmodels.tsa.statespace.sarimax import SARIMAX

model = SARIMAX(store\_sales['Sales'], order=(1,1,1), seasonal\_order=(1,1,1,7))

results = model.fit()

forecast = results.forecast(30)

**🔹 Option 3 : XGBoost / LightGBM (machine learning supervisé)**

python

CopierModifier

from xgboost import XGBRegressor

# Tu dois créer un dataframe avec des features temporelles (mois, jour, promo, etc.)

model = XGBRegressor()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

**🔹 Option 4 : LSTM (Deep Learning pour séries temporelles)**

Utilise TensorFlow ou PyTorch si tu veux explorer le deep learning.

**5. Évaluation des performances**

* Métriques : MAE, RMSE, MAPE

python

CopierModifier

from sklearn.metrics import mean\_absolute\_error

mae = mean\_absolute\_error(y\_test, y\_pred)

print("MAE :", mae)

**6. Visualisation interactive**

* Outils recommandés :
  + **Plotly** pour des graphiques interactifs
  + **Streamlit** ou **Dash** pour créer une app interactive avec sliders, filtres, etc.

python

CopierModifier

import plotly.express as px

fig = px.line(forecast, x='ds', y='yhat', title='Prévision des ventes')

fig.show()

**🧰 Résumé des bibliothèques utiles**

| **Fonction** | **Bibliothèques** |
| --- | --- |
| Analyse & prétraitement | pandas, numpy |
| Visualisation | matplotlib, seaborn, plotly |
| Modélisation | prophet, statsmodels, xgboost, sklearn, tensorflow |
| App interactive | streamlit, dash |

**🇨🇦 Jeu de données recommandé : Monthly Retail Trade Survey – Canada**

**🔗 Lien : https://open.canada.ca/data/en/dataset/d92b84cd-97ee-4a5c-bd24-0f73060b37cf**

[Retail Trade Sales by Province and Territory (Statistique Canada)](etapes.docx)

**🧾 Description du jeu de données :**

Ce jeu de données regroupe les **ventes mensuelles au détail** au Canada, ventilées par **province**, **territoire**, et **secteur d'activité (NAICS)**.

**✅ Données disponibles :**

* **Date** : au format AAAA-MM (par ex. 2020-05)
* **Province / territoire** : Ontario, Québec, Colombie-Britannique, etc.
* **Secteurs** : alimentation, essence, électronique, meubles, etc.
* **Ventes ($)** : en dollars canadiens (ajustés ou non pour la saisonnalité)

**🎯 Pourquoi ce jeu de données est pertinent :**

| **Critère** | **Présent ?** | **Commentaire** |
| --- | --- | --- |
| **Composante temporelle** | ✅ | Prévisions mensuelles possibles |
| **Granularité géographique** | ✅ | Analyse par province et comparaison |
| **Variables explicatives** | ✅ | Par secteur d’activité (effets sectoriels) |
| **Format simple (CSV)** | ✅ | Facile à charger dans Python |
| **Source fiable** | ✅ | Gouvernement du Canada (Statistique Canada) |

**💡 Ce que tu peux faire avec :**

**📈 Prévisions**

* Prévoir les ventes des provinces pour les mois à venir (ex. : Québec – commerce électronique)
* Comparaison entre modèles : Prophet, ARIMA, XGBoost, etc.

**📊 Visualisations interactives**

* Ventes dans le temps par province ou secteur avec **Plotly**
* Dashboard avec **Streamlit** : choix de la province, sliders temporels, graphiques interactifs

**🔍 Insights**

* Identifier les pics de ventes (effets saisonniers)
* Impact de crises économiques ou événements (COVID, inflation)

**🚀 Exemple de pipeline Python que tu pourrais construire :**

python

CopierModifier

import pandas as pd

import plotly.express as px

from prophet import Prophet

# 1. Charger les données

df = pd.read\_csv('retail\_sales\_canada.csv')

df['date'] = pd.to\_datetime(df['Ref\_Date'])

# 2. Filtrer Québec – Secteur : épiceries

qc\_data = df[(df['GEO'] == 'Quebec') & (df['NAICS'] == '4451')]

# 3. Format pour Prophet

df\_prophet = qc\_data[['date', 'VALUE']].rename(columns={'date': 'ds', 'VALUE': 'y'})

# 4. Modèle

model = Prophet()

model.fit(df\_prophet)

# 5. Prévisions

future = model.make\_future\_dataframe(periods=12, freq='M')

forecast = model.predict(future)

# 6. Visualisation interactive

fig = px.line(forecast, x='ds', y='yhat', title='Prévision des ventes au Québec - épiceries')

fig.show()