**Ministère de l’enseignement**

**Supérieur et de la recherche**

**UCAO-UUT**

### 

***INSTITUT SUPERIEUR DE TECHNOLOGIE ET D’INFORMATIQUE NUMERIQUE***

***LICENCE 3***

Présenté par:

AKLOBI kefas richard

Chargé du cours :

M. LOUKOUME AYOULA

## 1. Introduction

Ce projet a pour objectif de concevoir un système complet de gestion et d'analyse des ventes pour un magasin, en utilisant Python. Il repose sur la création d'une base de données de ventes simulées, l'analyse statistique de ces données et la visualisation des résultats à travers un tableau de bord interactif. L'ensemble est développé avec des bibliothèques modernes telles que pandas, matplotlib, seaborn, plotly et dash.

## 2. Modélisation de la base de données

Voici le schéma relationnel utilisé :

### PRODUITS ||--o{ VENTES : "1-N"

### CLIENTS ||--o{ VENTES : "1-N"

### PRODUITS {

### int id\_produit PK

### string nom\_produit

### string categorie

### float prix\_unitaire

### }

### CLIENTS {

### int id\_client PK

### string nom\_client

### string email

### }

### VENTES {

### int id\_vente PK

### date date\_vente

### int quantite

### float montant\_total

### }

### Code SQL de création :

# Création des tables

        c.execute('''CREATE TABLE IF NOT EXISTS produits

                    (id INTEGER PRIMARY KEY,

                     nom TEXT NOT NULL,

                     categorie TEXT CHECK(categorie IN ('Informatique', 'Mobile', 'Audio', 'Maison', 'Bureau')),

                     prix\_unitaire REAL NOT NULL CHECK(prix\_unitaire > 0))''')

        c.execute('''CREATE TABLE IF NOT EXISTS clients

                    (id INTEGER PRIMARY KEY,

                     nom TEXT NOT NULL,

                     email TEXT UNIQUE,

                     ville TEXT)''')

        c.execute('''CREATE TABLE IF NOT EXISTS ventes

                    (id INTEGER PRIMARY KEY,

                     produit\_id INTEGER NOT NULL,

                     client\_id INTEGER NOT NULL,

                     date TEXT NOT NULL,

                     quantite INTEGER NOT NULL CHECK(quantite > 0),

                     montant REAL NOT NULL,

                     FOREIGN KEY(produit\_id) REFERENCES produits(id),

                     FOREIGN KEY(client\_id) REFERENCES clients(id))''')

## 3. Insertion et génération des données

### Script : create\_db.py

### 10 produits , 20 clients, 200 ventes aléatoire

### Extrait de code :

Commentaires :

* Le montant total est calculé dynamiquement.
* La date est générée sur les 12 derniers mois.

## 4. Extraction des données

Via extract\_data.py, les données sont extraites avec jointures via pandas :

### Extrait :

 # Insertion de produits

        produits = [

            (1, "Laptop Elite", "Informatique", 999.99),

            (2, "Smartphone Pro", "Mobile", 799.99),

            (3, "Casque Audio", "Audio", 149.99),

            (4, "Clavier Mécanique", "Informatique", 89.99),

            (5, "Souris Gaming", "Informatique", 59.99),

            (6, "Enceinte Bluetooth", "Audio", 129.99),

            (7, "Lampe LED", "Maison", 39.99),

            (8, "Cahier Premium", "Bureau", 19.99),

            (9, "Stylo 3D", "Bureau", 29.99),

            (10, "Tapis de Souris", "Informatique", 24.99)

        ]

        c.executemany("INSERT INTO produits VALUES (?,?,?,?)", produits)

        # Insertion de clients

        clients = [(i, fake.name(), fake.unique.email(), fake.city()) for i in range(1, 21)]

        c.executemany("INSERT INTO clients VALUES (?,?,?,?)", clients)

## 5. Visualisations

## Les résultats des analyses sont visualisés à l'aide de graphiques générés dans le notebook. Par exemple, le graphique ca\_par\_mois.png illustre le chiffre d'affaires mensuel, mettant en évidence des pics de ventes en décembre, probablement dus aux fêtes de fin d'année.

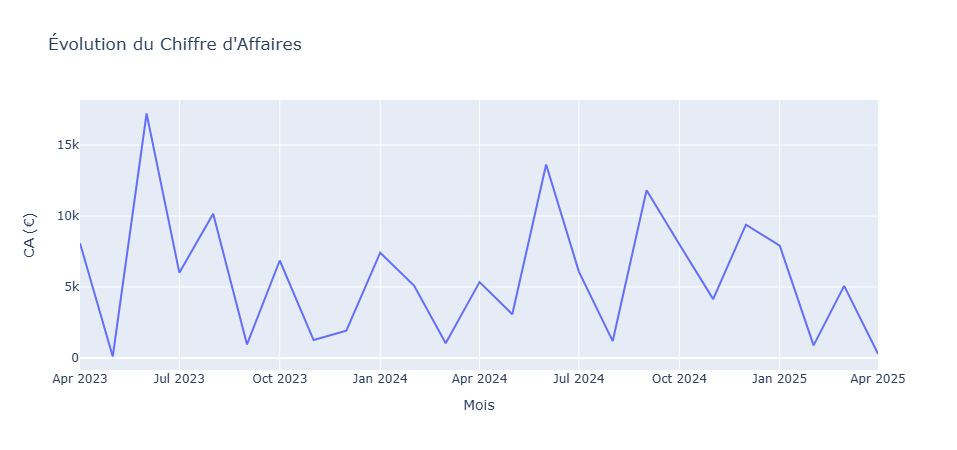
## 6. Tableau de bord interactif avec Streamlit

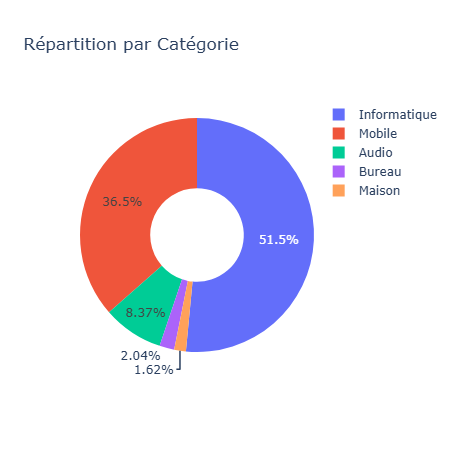
Un tableau de bord interactif est développé avec Streamlit via le script dashboard\_ventes.py. Ce tableau de bord offre une interface utilisateur intuitive pour explorer les données de vente, avec des filtres par catégorie de produit et par période. Les indicateurs clés affichés incluent le chiffre d'affaires total, le nombre de transactions et le panier moyen.

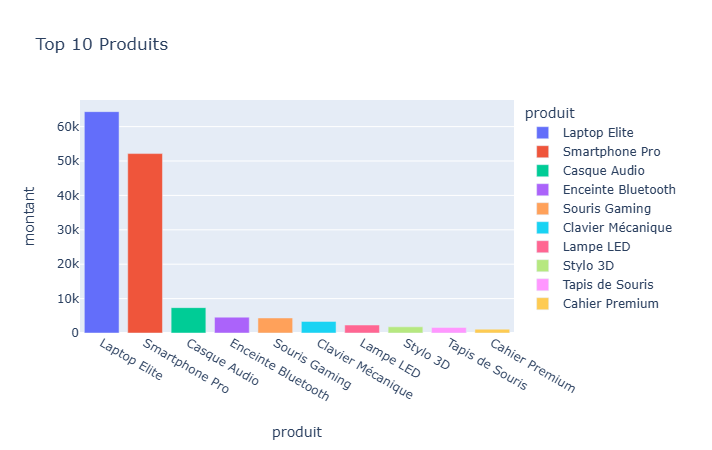
7. Conclusion

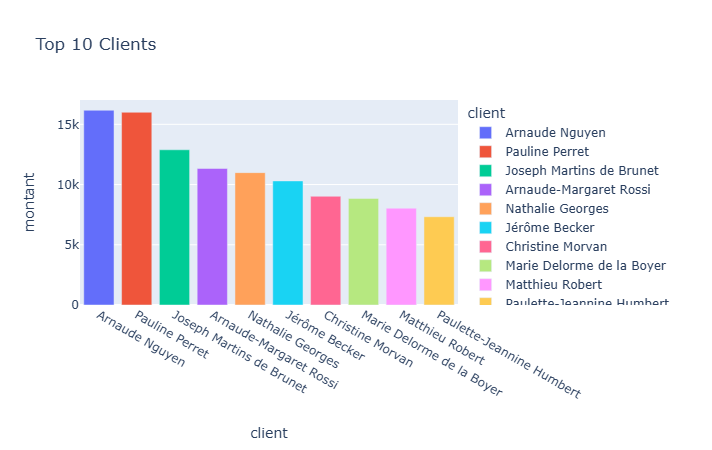
Ce projet démontre la capacité à concevoir une solution complète pour la gestion des ventes, depuis la modélisation de la base de données jusqu'à la création d'un tableau de bord interactif. Les outils développés fournissent des insights précieux pour optimiser les performances commerciales du magasin.

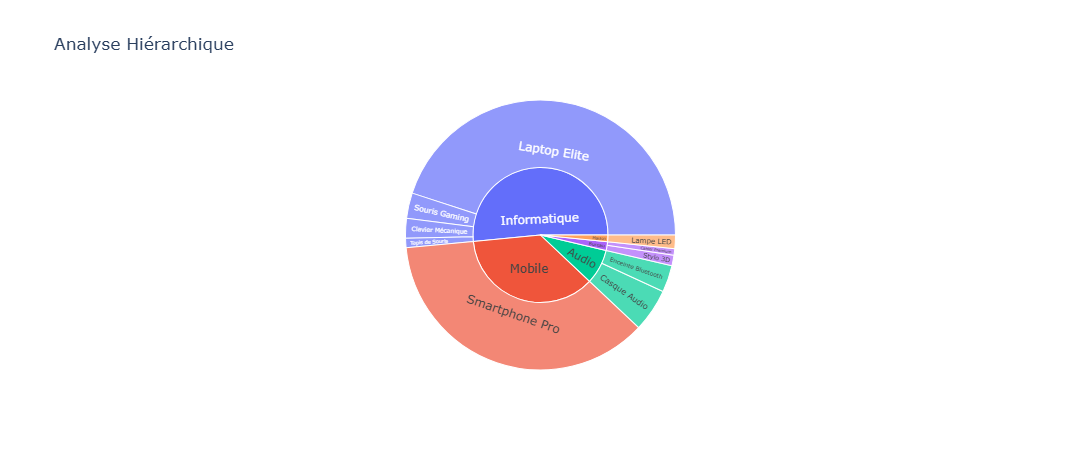
8. Captures du dashboard

****

****







# Clonez le dépôt

git clone (https://github.com/Kefas-richard/projet-de-gestion-de-vente)

# Installez les dépendances

pip install -r requirements.txt

# Générer la base de données

python scripts/create\_db.py

# lancer l'analyse

python scripts/analyse\_ventes.py

# Lancer le dashboard

python scripts/dashboard.py