**📘 Rapport de Projet**

**Analyse des ventes d’un magasin Projet Python**

**Filière** : Génie Informatique/ESI

**Parcours** : Licence 3

**Nom** : AGBO Komi Ludovic

**Encadrant** : M. LOUKOUME

**Date** : 08/04/2025

**Table des Matières**

**Introduction  
Présentation du Projet General  
Architecture Technique du Projet**

**Modélisation de la Base de Données**

**Génération Automatique des Données**

**Analyse Statistique des Données**

**Visualisation des Données**

**Problèmes Rencontrés et Résolutions**

**Introduction**

L'objectif principal de ce projet est de développer un système automatisé permettant d'analyser les ventes d'un magasin. Le système comprend la génération de données réalistes, leur stockage dans une base SQLite, leur traitement via Python et une représentation graphique des résultats.

Dans ce contexte, j’ai réalisé un pipeline complet de bout en bout allant de la conception de la base de données à la production d’un dashboard interactif. Ce rapport détaille toutes les étapes du projet, les outils utilisés, les problèmes rencontrés et les perspectives d'amélioration.

**Présentation Générale du Projet**

Le projet repose sur les éléments suivants :

* Une **base de données relationnelle** SQLite
* Un système de **génération automatique de données**
* Des **analyses statistiques** : chiffre d’affaires, panier moyen, meilleurs produits et clients
* Des **visualisations graphiques** (statiques en PNG et interactives en HTML)
* Une **organisation en modules Python**, pour assurer clarté et maintenabilité

**Architecture Technique du Projet**

**📁 Arborescence du Projet**

projet-ventes/

├── .venv/ # Environnement virtuel Python

├── database/ # Contient la base de données SQLite (ventes\_magasin.db)

├── documentation/ # Contient le fichier de documentation du projet

├── rapport/ # Résultats finaux

└── dashboard\_ventes.html # Dashboard interactif HTML généré

├── scripts/ # Scripts Python principaux

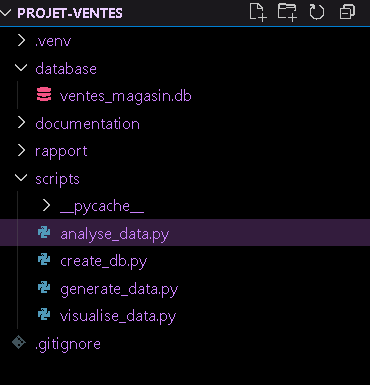
├── create\_db.py # Crée la base de données avec les tables

├── generate\_data.py # Génère des données fictives avec Faker

├── analyse\_data.py # Fait les calculs : CA, panier moyen, etc.

└── visualize\_data.py # Génère les graphiques et le dashboard

📸 **Capture à insérer :** explorateur de fichiers affichant l’arborescence.



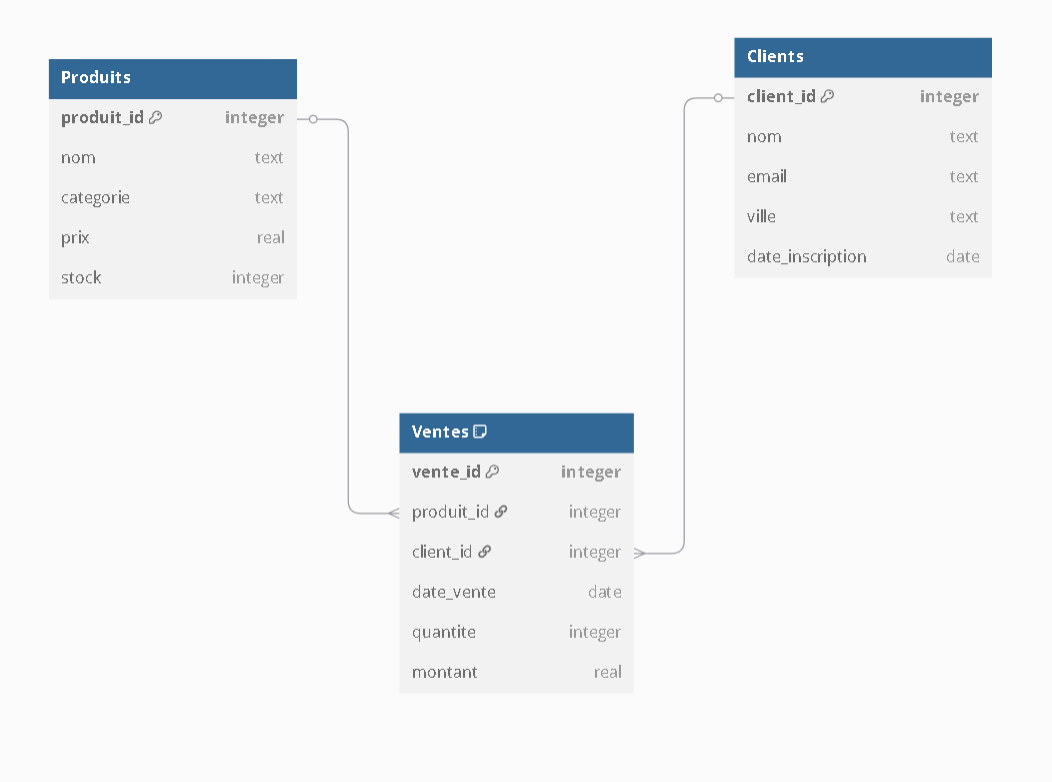
**Modélisation de la Base de Données**

Trois tables ont été créées pour simuler les opérations commerciales d’un magasin :

* Produits (produit\_id, nom, categorie, prix, stock)
* Clients (client\_id, nom, email, ville, date\_inscription)
* Ventes (vente\_id, produit\_id, client\_id, date\_vente, quantite, montant)

Cette structure permet de relier chaque vente à un produit et à un client, tout en gardant des informations importantes pour les analyses (stock, ville, dates, etc.).

📸 **Capture à insérer :** schéma relationnel



**Génération Automatique des Données**

Le script generate\_data.py utilise la bibliothèque Faker pour générer :

* 100 produits répartis en plusieurs catégories (alimentaire, électronique, etc.)
* 50 clients avec des villes variées
* Environ 1000 ventes réparties sur une année

Un point d’attention important a été la cohérence des données (prix, quantités, montants), et le respect du format EUR pour éviter les problèmes de conversion.

📸 **Captures à insérer :**

* Extrait de code de generate\_data.py
* Données générées affichées dans la base

**Analyse Statistique des Données**

À l’aide de pandas, les analyses suivantes ont été réalisées dans analyse\_data.py :

* 💰 **Chiffre d’affaires total** (somme des montants des ventes)
* 🛒 **Panier moyen** (montant moyen par commande)
* 📦 **Top 5 des produits** les plus vendus
* 👤 **Top 5 des clients** les plus dépensiers
* 📊 **Évolution mensuelle du CA**

Des fonctions spécifiques ont été créées pour chaque analyse, avec retour en console ou fichier.

📸 **Captures à insérer :**

* Code de calculs statistiques
* Résultats imprimés dans le terminal

**9. Visualisation des Données**

Deux types de visualisation ont été produits :

**🔹 Graphiques Statique (PNG – matplotlib)**

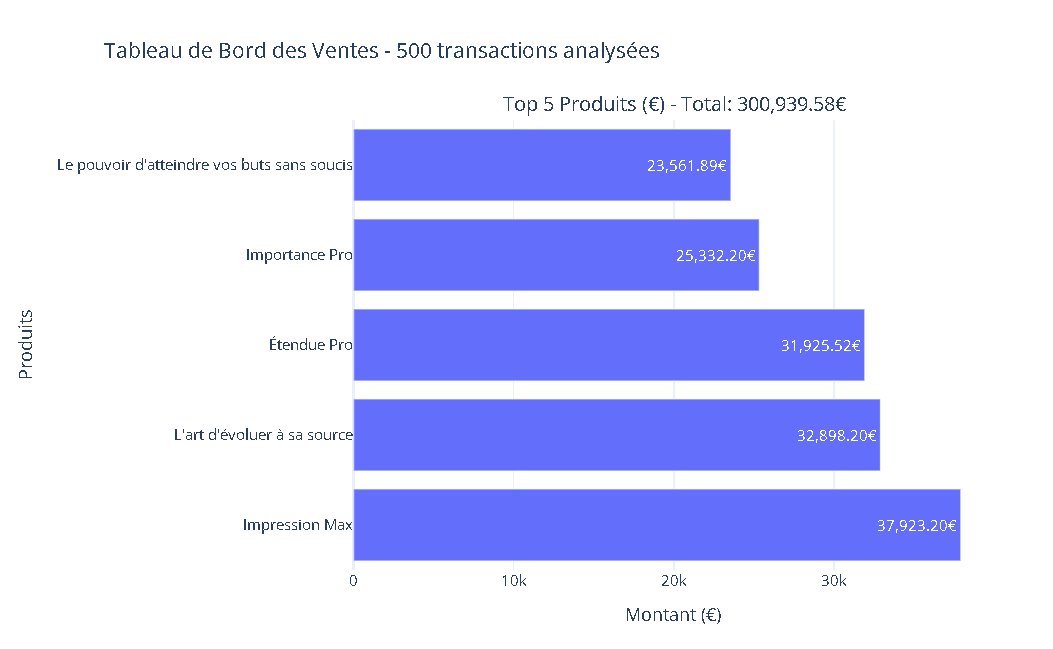
* Histogrammes : top produits, top clients
* Courbes : évolution du chiffre d’affaires par mois

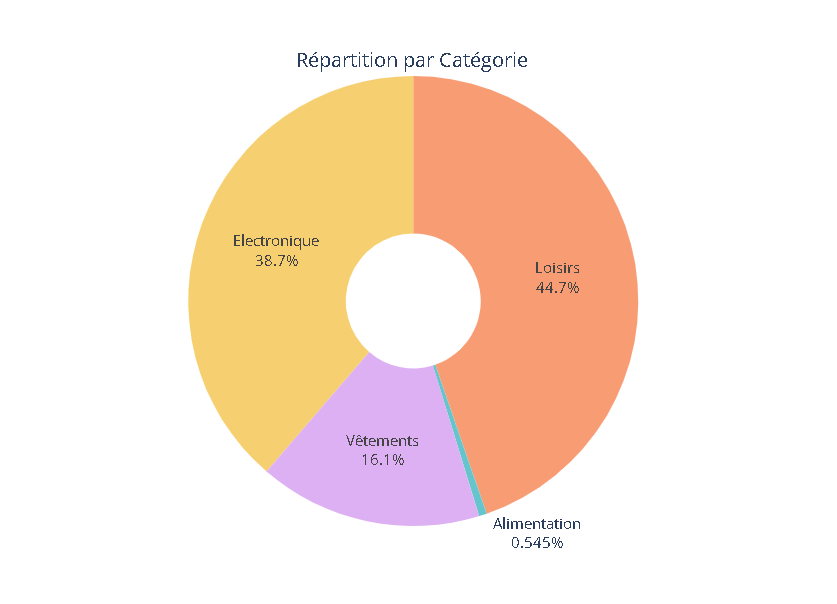
**🔹 Dashboard (HTML – plotly)**

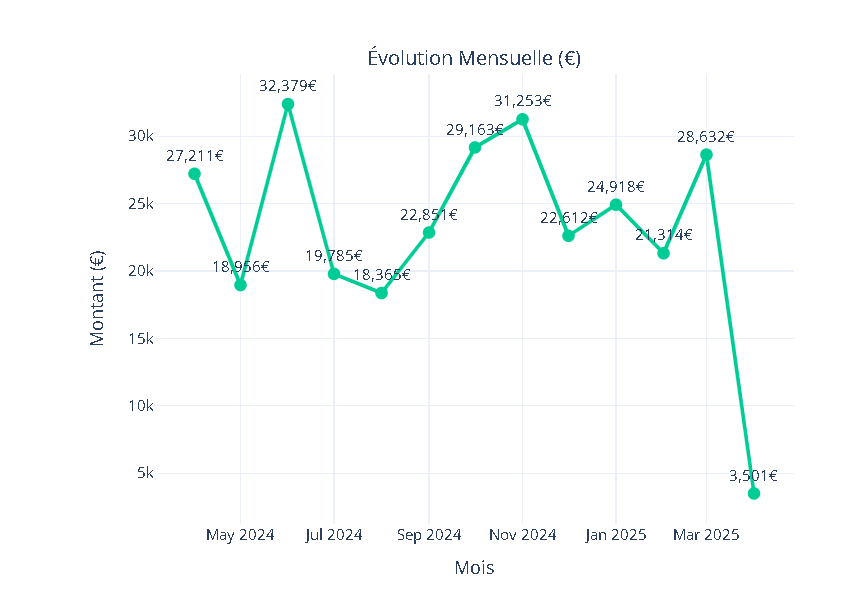
* Consultable depuis un navigateur
* Permet un survol pour afficher les données précises

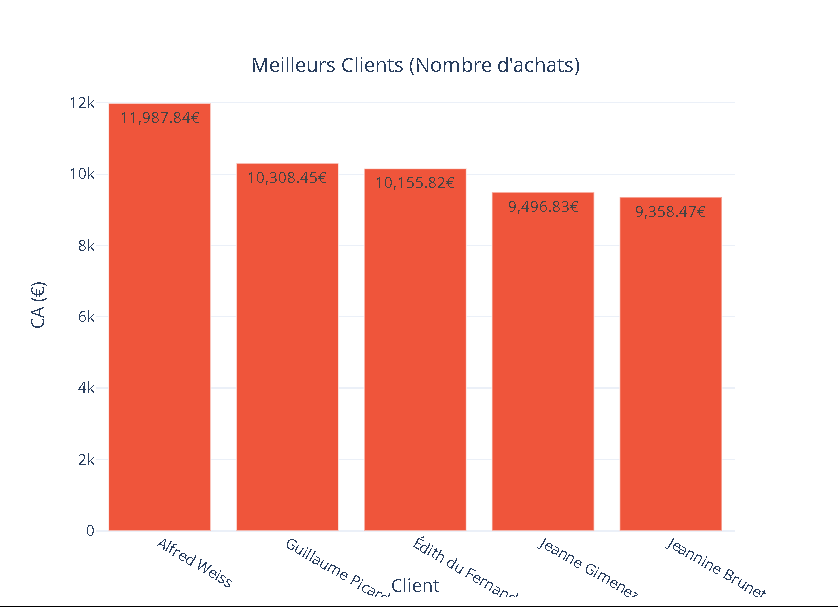
📸 **Captures à insérer :**

* Graphiques PNG générés









* Dashboard ouvert dans un navigateur



**Problèmes Rencontrés et Résolutions**

**🧱 Version initiale**

* Devise mixte EUR et FCFA : *erreurs de conversion fréquentes*
* Visualisations incohérentes selon la devise
* Complexité inutile à ce stade

**🔧 Version finale**

* Suppression complète du FCFA
* Unification du système en EUR
* Correction des erreurs de type (float vs int)
* Requêtes SQL optimisées
* Dashboard réécrit avec format standardisé

**Fonctionnalités Clés Implémentées**

|  |  |  |
| --- | --- | --- |
| **Fonctionnalité** | **Implémentée ?** | **Commentaire** |
| Génération données | ✅ | Faker + fonctions personnalisées |
| CA, panier moyen | ✅ | Calculs précis dans analyse\_data.py |
| Visualisations statiques | ✅ | matplotlib |
| Dashboard | ✅ | plotly + export HTML |
|  |  |  |

**Conclusion**

Ce projet a permis de mettre en œuvre une approche professionnelle et modulaire de l’analyse de données commerciales. À partir de données simulées, il a été possible de construire une solution complète de pilotage des ventes.

J’ai ainsi pu renforcer mes compétences sur :

* Les bases de données relationnelles (SQLite)
* La manipulation de données avec pandas
* La visualisation graphique avec matplotlib et plotly
* La modularisation en Python

**Annexes**

* 📁 scripts/\*.py : fichiers sources
* 📁 ventes.db : base de données finale
* 📁 output/\*.png : graphiques
* 📄 dashboard\_ventes.html : tableau de bord
* 🧰 Libs utilisées : pandas, sqlite3, faker, plotly, matplotlib