

Filière : Ingénierie Numérique en Data science,  
Intelligence Artificielle Santé Digitale

## MINI PROJET

---

# Collecte et Analyse de Données d'un Site Web E-Commerce

---

Réalisé par:

ER-REGHAY Zaynab.  
HASNAOUI Fatima ezzahra

Année Universitaire: 2024-2025

# Contents

|             |  |          |
|-------------|--|----------|
| <b>red1</b> | <b>Exploratory Data Analysis (EDA)</b>     | <b>2</b> |
| red1.1      | Notebook EDA . . . . .                     | 2        |
| red1.2      | Notebook Faker . . . . .                   | 2        |
| red1.3      | Notebook EDA avancée . . . . .             | 2        |
| red1.4      | Notebook Regression . . . . .              | 2        |
| <b>red2</b> | <b>Introduction</b>                        | <b>2</b> |
| <b>red3</b> | <b>Fonctionnalités de l'application</b>    | <b>3</b> |
| red3.1      | Fonctionnalité de prédiction . . . . .     | 3        |
| red3.2      | Statistiques descriptives . . . . .        | 3        |
| red3.3      | Visualisation . . . . .                    | 3        |
| red3.4      | Intégration Power BI . . . . .             | 3        |
| <b>red4</b> | <b>Structure du code</b>                   | <b>4</b> |
| red4.1      | Structure des dossiers du projet . . . . . | 4        |
| red4.2      | Routes Flask . . . . .                     | 4        |
| red4.3      | Workflow de prédiction . . . . .           | 4        |
| <b>red5</b> | <b>Technologies utilisées</b>              | <b>5</b> |
| <b>red6</b> | <b>Détails de l'application</b>            | <b>5</b> |
| <b>red7</b> | <b>Conclusion</b>                          | <b>5</b> |

# 1 Exploratory Data Analysis (EDA)

Avant de développer l'application, une série de notebooks Jupyter a été utilisée pour analyser et préparer les données. Voici un résumé des quatre notebooks utilisés :

## 1.1 Notebook EDA

Le premier notebook a permis une analyse exploratoire de base du jeu de données. Il a révélé les caractéristiques essentielles des données, identifié les colonnes pertinentes, et effectué un premier nettoyage des données.

## 1.2 Notebook Faker

Le deuxième notebook a utilisé la bibliothèque Faker pour ajouter des colonnes supplémentaires au jeu de données, enrichissant ainsi le dataset avec des données de simulation pour enrichir l'analyse et tester différents modèles.

## 1.3 Notebook EDA avancée

Le troisième notebook a été dédié à une analyse plus approfondie des données. Il a généré diverses visualisations, dont celles utilisées dans l'application Flask pour la répartition par âge, la répartition par catégorie et la segmentation des utilisateurs.

## 1.4 Notebook Regression

Le quatrième notebook a permis de générer le modèle de prédiction stocké dans '`stock_prediction_model.pkl`'.

Ce notebook a utilisé les données transformées pour entraîner plusieurs modèles d'apprentissage automatique, y compris des modèles linéaires et non linéaires. Après l'entraînement, il a permis de sélectionner le modèle le plus performant en fonction de critères tels que l'accuracy, le f1-score, et la robustesse au surapprentissage. Le modèle final a été enregistré pour une utilisation future dans l'application Flask, permettant des prédictions efficaces et rapides des besoins en stock.

## 2 Introduction

Ce document présente une vue d'ensemble de l'application Flask développée pour la prédiction des stocks et l'analyse des données. Les fonctionnalités de l'application incluent :

- Prédire les besoins futurs en stock en fonction des entrées utilisateur.
- Afficher des statistiques descriptives.
- Visualisation des données (par exemple, répartition par âge, répartition par catégorie).
- Intégration avec un tableau de bord Power BI.

## 3 Fonctionnalités de l'application

### 3.1 Fonctionnalité de prédiction

L'application prédit les besoins futurs en stock en utilisant un modèle d'apprentissage automatique pré-entraîné. Le formulaire de prédiction est implémenté sur une page HTML dédiée avec des champs conviviaux pour les entrées.

**Entrées :**

- Quantité vendue
- Prix unitaire
- Âge du client
- Stock actuel
- Évaluation du produit
- Jours depuis l'inscription

**Sorties :**

- Besoin en stock prédit (affiché en unités).

## 3.2 Statistiques descriptives

Une page dédiée affiche des statistiques descriptives du jeu de données en utilisant la bibliothèque `pandas`. Ces statistiques incluent la moyenne, la médiane, l'écart-type, et d'autres mesures.

## 3.3 Visualisation

L'application affiche des visualisations pré-générées :

- Répartition par âge
- Répartition par catégorie de produit
- Segmentation des utilisateurs

Ces images sont intégrées dans l'interface web et accessibles via des routes spécifiques.

## 3.4 Intégration Power BI

Un tableau de bord Power BI est intégré dans l'application via un `iframe`. Ce tableau offre une exploration avancée des données et des analyses visuelles.

# 4 Structure du code

## 4.1 Structure des dossiers du projet

Voici la structure du projet Flask :

## 4.2 Routes Flask

L'application utilise les routes suivantes :

- `/` - Page d'accueil.
- `/statistics` - Statistiques descriptives.

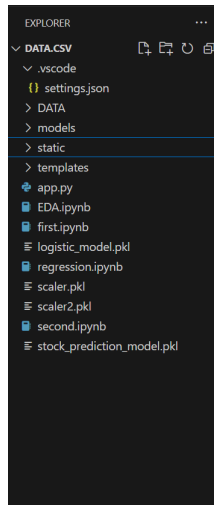


Figure 1: Capture d'écran de la structure des dossiers du projet

- `/predict` - Formulaire de prédiction et résultat.
- `/age_distribution` - Visualisation de la répartition par âge.
- `/category_distribution` - Visualisation des catégories de produits.
- `/user_segmentation` - Visualisation de la segmentation des utilisateurs.
- `/power_bi_dashboard` - Tableau de bord Power BI intégré.

### 4.3 Workflow de prédiction

Le processus de prédiction suit les étapes suivantes :

1. Accepter les entrées utilisateur via un formulaire HTML.
2. Normaliser les entrées à l'aide d'un scaler pré-entraîné (`scaler.pkl`).
3. Prédire le besoin en stock à l'aide d'un modèle pré-entraîné (`stock_prediction_model.pkl`).
4. Afficher le résultat sur la page de prédiction.

## 5 Technologies utilisées

- **Bibliothèques Python** : Flask, pandas, scikit-learn, joblib, matplotlib, seaborn.
- **Frontend** : HTML, CSS (styles inline pour certaines pages).
- **Intégration** : Power BI.
- **Outils** : Modèles d'apprentissage automatique pré-entraînés, visualisations pré-générées.

## 6 Détails de l'application

Cette application est conçue pour simplifier la prédiction des stocks et fournir des insights complets sur les données via des interfaces conviviales et des visualisations. Chaque fonctionnalité est développée avec une attention particulière à l'évolutivité et à l'utilisabilité. L'intégration avec Power BI ajoute de la valeur en permettant des analyses et rapports en temps réel.

## 7 Conclusion

Cette application Flask combine apprentissage automatique, analyse de données, et visualisation pour fournir un outil complet de prédiction des stocks et d'exploration des données. Les améliorations futures pourraient inclure l'intégration de données en temps réel et l'élargissement des capacités d'analyse.