🎯 Projet DVF : Scraping & Analyse immobilière

Ernest Loïc ENGOUE et Maryline FONTA

PMN 2025 Mastère Data Engineer parcours Data Analyst DAN 25.1 – 29/08/2025

Objectif du projet

Télécharger et nettoyer les données DVF (valeurs foncières) depuis **data.gouv.fr**

Construire un pipeline automatisé :

Scrapy → Nettoyage → Dashboard Streamlit

Permettre une visualisation claire des transactions immobilières

Architecture du pipeline

Scrapy Spider → CSV.gz DVF → Nettoyage (Python/Pandas) → Fichier propre → Github (CI/CD) → Dashboard Streamlit

L’intégration continue (CI/CD) via GitHub Actions permet d’automatiser le téléchargement, le nettoyage et la mise à jour régulière des données.

Etape 1 : Scrapy Spider.py

Téléchargement des fichiers .csv.gz par département et année

Décompression et parsing CSV

Extraction des champs utiles : prix, surface, type de bien, localisation…

Export automatique en JSON

Etape 2 : Nettoyage cleaner.py

Filtrage : uniquement biens avec prix + surface
Uniformisation des adresses
Calcul du prix moyen au m²

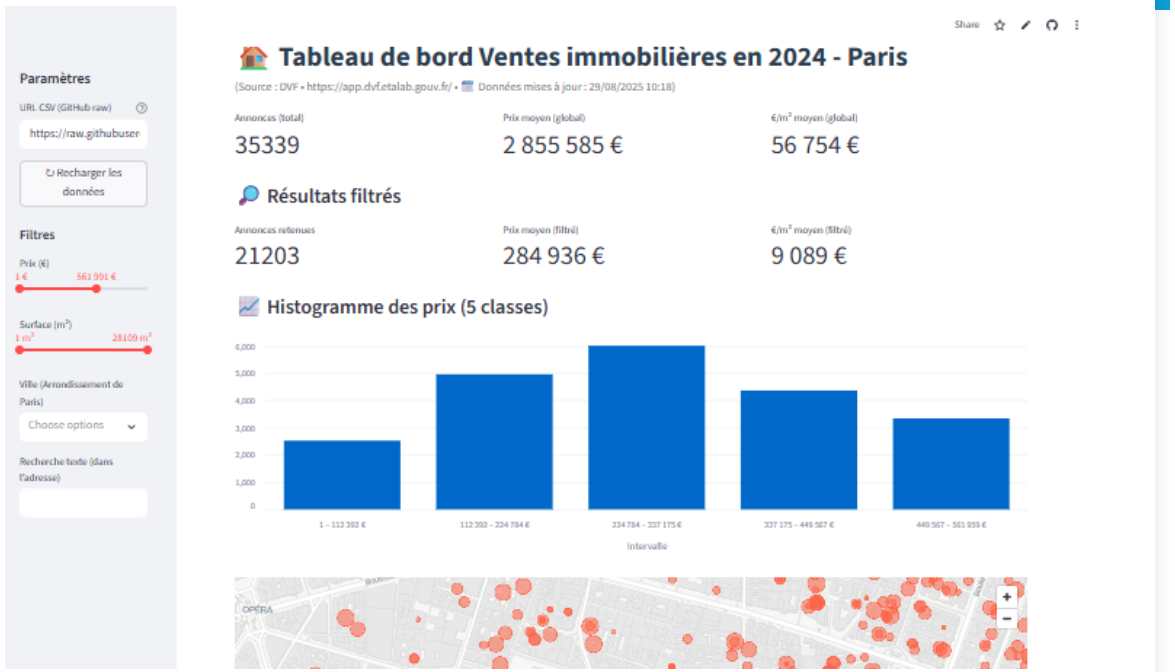
Etape 3 : Dashboard streamlit app.py

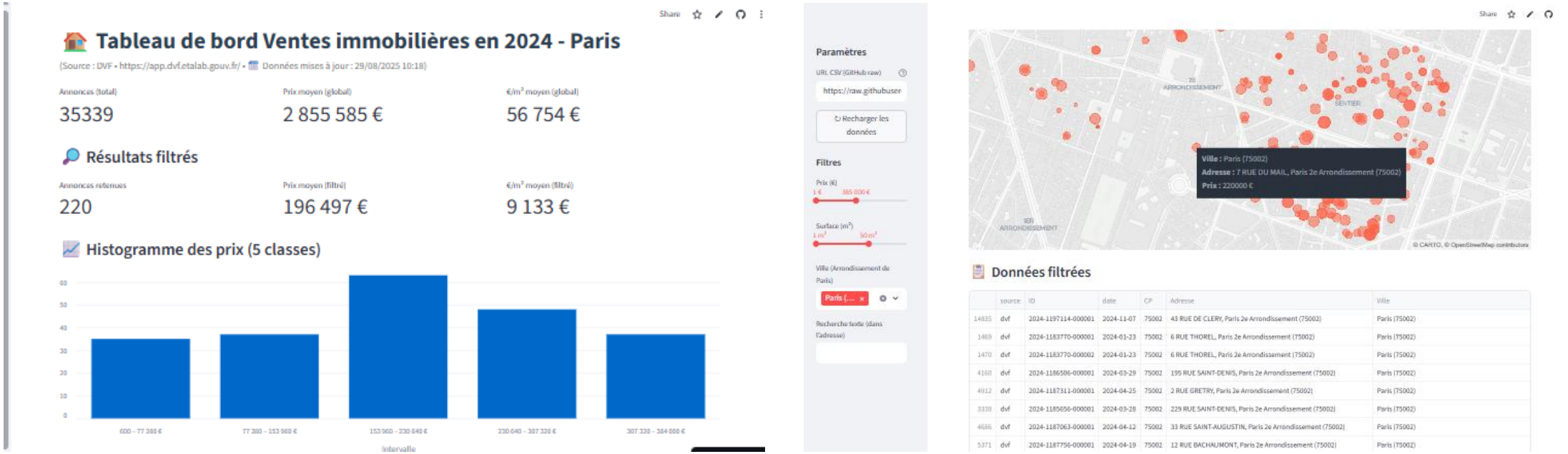
Upload du fichier nettoyé

Visualisation interactive :

- prix moyens
- distribution des surfaces
- carte géographique des ventes

Interface simple et interactive avec filtres





Etape 3 : Dashboard streamlit

Etape 4 : Mise en place CI/CD avec GitHub Actions

Création d'un fichier .github/workflows/main.yml

```yaml
name: Scrape & Clean (CI)

on:
  workflow_dispatch:        # lancement manuel
  push:
    branches: [ "main" ]

permissions:
  contents: write           # nécessaire pour commit/push avec GITHUB_TOKEN

concurrency:
  group: scrape-clean
  cancel-in-progress: true
```

Défis rencontrés

- Blocages potentiels (robots.txt)

  Accès facile aux données **ouvertes DVF** (data.gouv.fr) **mais blocage sur SeLoger** :

  protections anti-scraping

  cadre légal différent (pas de licence ouverte)

- Nettoyage du CSV (format variable, encodage, décimales FR)

  Format de la ville avec arrondissement

- Gestion des fichiers volumineux (.csv.gz) → blocage de github

Gestion de projet

Séparation des étapes(Scrapy / Cleaning / GitHub / Dashboard)

Automatisation avec un pipeline

GitHub pour versionner et centraliser le code :

- Mise en place de workflows GitHub Actions pour lancer automatiquement le pipeline à chaque mise à jour et déployer le dashboard en continu.

Documentation avec README

Améliorations possibles

Sources de données :

enrichir avec d'autres jeux de données (France entière, cadastre, INSEE, DPE…).

Dashboard :

ajouter plus de filtres interactifs

comparaisons entre départements

tendances temporelles

Déploiement :

Intégrer le streamlit dans une page web d'un site.

Leçons apprises

Scrapy simplifie et sécurise le scraping & l'export

Le nettoyage représente la plus grosse part du travail.

Importance du contrôle de version : on garde un historique des changements, on peut revenir en arrière, et c'est indispensable en projet réel.

Streamlit permet une valorisation rapide et efficace des données.

Gestion de projet et choix des sources sont essentiels.

Nécessité de documenter et planifier dès le début

Défi : accès aux données seloger

Nous avions commencé notre projet en nous appuyant sur les données du site **SeLoger**, dans l'idée d'extraire et d'analyser les annonces immobilières.

Cependant, nous avons été confrontés à un problème de cadre légal, car les données de SeLoger ne sont pas publiées en open data et leur extraction automatisée n'est pas autorisée.

Face à cette contrainte, nous avons choisi de réorienter notre travail vers les données DVF, mises à disposition par l'État en open data.

Ce changement nous a permis de rester dans un cadre légal clair tout en travaillant sur un jeu de données riche, fiable et parfaitement adapté à notre projet.

Conclusion

Projet = **pipeline complet automatisé**

Données DVF accessibles et visualisables simplement

Compétences acquises : scraping, data cleaning, visualisation, gestion de projet

Un projet qui allie technique et gestion de données.