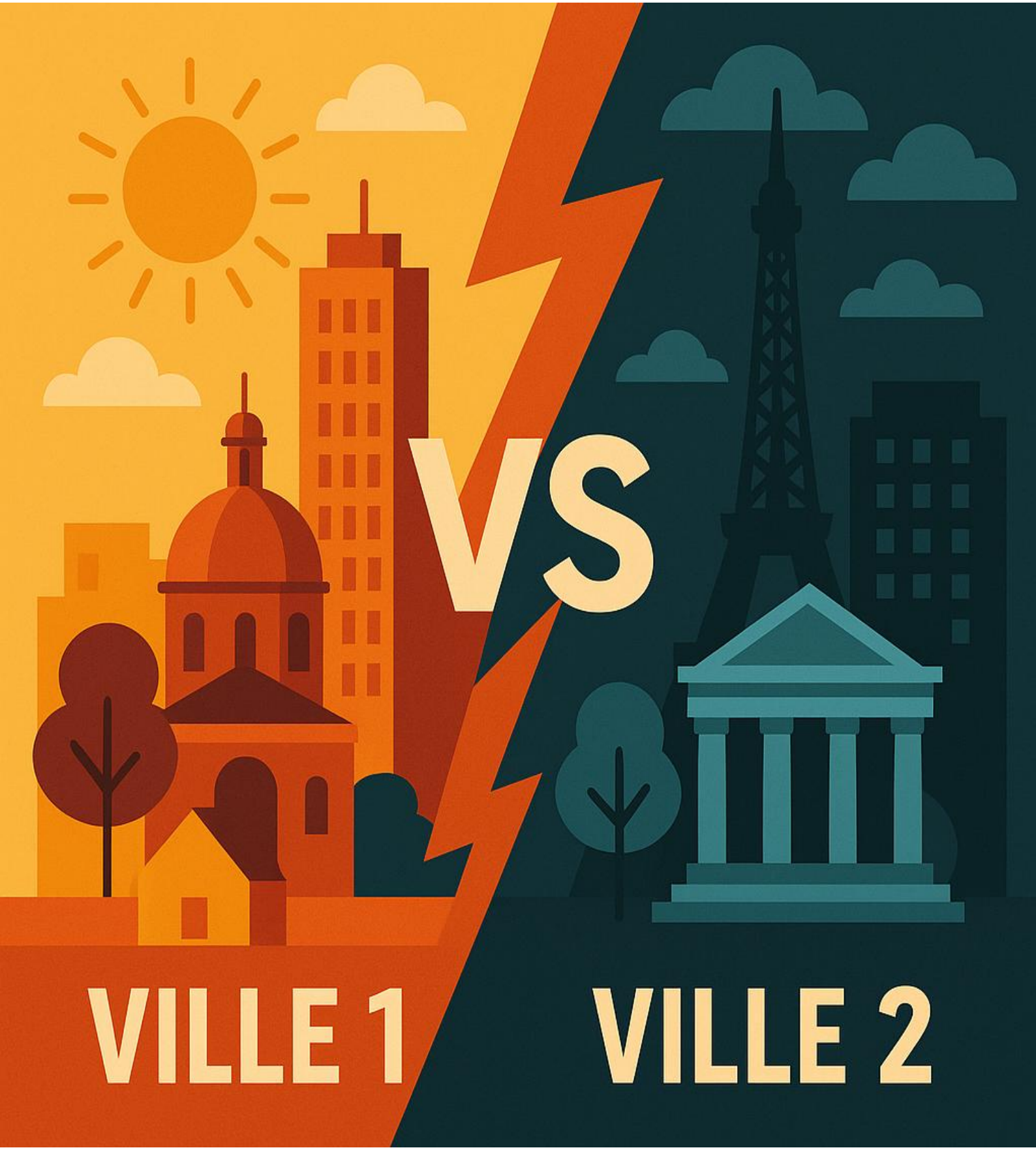
NOM: BASTIEN EBELY | Mehdi BENAYED

PROJET Dashboard City Fighting



Année universitaire : 2024 – 2025

Groupe 34

2. Table des Matières

3. Introduction

Contexte du projet SAE

Dans le cadre du cours *SAE Outils Décisionnels*, nous avons été amenés à développer une application web interactive. Celle-ci vise à mettre en œuvre des compétences en traitement de données, en visualisation et en développement web à travers un cas d'usage concret.

Objectif de l'application

L'application "Dashboard City Fighting" permet la comparaison entre deux villes françaises de plus de 20 000 habitants, selon le choix que l'utilisateur pourra choisir de comparer les performances. Afin de mettre en pratique la collecte, l'analyse et la visualisation de données à partir de sources ouvertes (open data, APIs...). L'application visera à aider à la prise de décision : par exemple dans le cadre où une personne souhaite déménager, investir, ou tout simplement mieux comprendre les différences entre deux territoires. Cette application aidera l'utilisateur un maximum.

Présentation des aspects analysés

L'application met en avant des données concrètes tel que :

- Présentation des données générales (population, superficie, etc.)
- Logement (loyers moyens, types de logements...)
- Météo (climat annuel et météo à court terme)

4. Présentation de l'Interface Web

Technologie utilisée

L'application est développée avec Streamlit, permettant une mise en page simple et réactive orientée utilisateur.

Adresse de l'application

Streamlit

Fonctionnalités principales

- Sélection de deux villes parmi une liste et comparer sur plusieurs critères
- Comparaison dynamique avec visualisation graphiques
- Filtres pour affiner les critères analysés
- Cartes, graphiques, indicateurs visuels

5. Sources de Données et API

Nous avons 4 API que nous avons utilisé dans le cadre de ce projet qui sont :

1) API Overpass (OpenStreetMap) :

- Utilisée pour récupérer les limites de la commune et les points d'intérêt (POI) autour d'une ville via la fonction get_commune_boundary et get_pois_from_overpass.
- URL de l'API : http://overpass-api.de/api/interpreter

2) API Geo (France) :

- Utilisée pour obtenir des informations géographiques détaillées sur les communes (code INSEE, population, surface, centre géographique, etc.).
- URL de l'API : https://geo.api.gouv.fr/communes

3) API Open-Meteo:

- Utilisée pour obtenir des informations météorologiques actuelles et des prévisions pour les communes.
- URL de l'API : https://api.open-meteo.com/v1/forecast

4) API logement (fichiers CSV locaux) :

- Bien que ce ne soit pas une API distante, tu charges des fichiers CSV qui contiennent des informations sur le logement pour les communes, et ces données sont utilisées pour récupérer des informations sur les maisons et appartements dans les villes.
- Source : Fichiers locaux comme api_logement_2014.csv, api_logement_2015.csv, etc.

Donc :

- Deux API externes pour géolocalisation et points d'intérêt (overpass-api.de et geo.api.gouv.fr).
- Une API météorologique (open-meteo.com).
- Des fichiers CSV locaux pour les données de logement.

Données Générales de la population

Ces informations sont issues de l'API gouvernementale. Elles permettent une première lecture du territoire :

- Source : INSEE
- Type : Fichier CSV
- Intégration : Chargement manuel
- Traitement : Filtrage sur les villes > 20 000 habitants
- Nom de la ville
- Population
- Superficie (en km²)
- Densité de population (hab/km²) : calculée automatiquement en divisant la population par la superficie

Ces indicateurs offrent une vision structurelle de la ville et permettent d'évaluer son échelle et sa densité.

Météo

L'application interroge l'API Meteo pour récupérer :

- *La température actuelle (en °C)*
- *La vitesse du vent actuelle*
- *Un tableau de prévisions à 7 jours avec :*
    - Température minimale et maximale quotidienne
    - Précipitations prévues (en mm)

Cette donnée est utile pour les utilisateurs qui souhaitent comparer la qualité de vie ou le climat dans un quartier de villes précises.

Logement

À partir de fichiers CSV fournis (api_logement_2014.csv → 2023), l'application extrait les données immobilières les plus récentes (année 2023 dans les exemples) via le code INSEE de chaque commune :

- *Nombre de maisons vendues*
- *Nombre d'appartements vendus*
- *Prix moyen de vente (€)*
- *Prix au mètre carré (€ / m²)*
- *Surface moyenne des biens vendus (m²)*

Cela permet une lecture rapide de la pression immobilière et du niveau d'accessibilité au logement dans la ville.

5.1 Zoom sur les API

Exemple : API Wikipédia

- Choisie pour sa facilité d'accès et ses mises à jour fréquentes
- Difficultés : Parsing des résultats HTML, données peu structurées
- Solutions : Utilisation de bibliothèques de nettoyage
- Alternative envisagée : OpenDataSoft (trop limitée pour les critères visés)

Catégorie	Source	Type de données
Données générales	geo.api.gouv.fr	INSEE, surface, densité
Météo	open-meteo.com + Overpass OpenStreetMap	Température, vent, prévisions
Logement	Fichier CSV locaux	Prix, surface, ventes par type

6. Déploiement de l'Application

Plateforme choisie

- Streamlit Cloud (streamlit.io)

Raisons du choix

- Intégration directe avec GitHub
- Facilité de déploiement et d'actualisation

Étapes du déploiement

- Création du dépôt GitHub

- Configuration du fichier requirements.txt
- Connexion à Streamlit Cloud
- Build automatique

Problèmes rencontrés

- Gestion des dépendances Python (pandas, requests…) + création de l'environnement virtuel sur le Z.
- Limites de quota API sur certaines plateformes (le token de l'API du site gouvernemental s'expire à partir de 2 semaines).

7. Mode d'Emploi

Lancement en local

Prérequis : version Python par défault, pip install -r requirements.txt
Commande : streamlit run app.py

Lancement en ligne

Accès via l'adresse publique du déploiement.

Navigation

1. Sélectionner deux villes
2. Cliquer sur "Comparer"
3. Visualiser les indicateurs et interpréter les résultats pour une personne souhaitant investir ou déménager dans une ville précise

8. Scripts et Fichiers du Projet

Contenu de l'archive

- app.py : script principal
- data_loader.py : gestion des sources de données

GitHub

Shadow75016/appville

Mehdi-In-Coding

9. Conclusion

Bilan du projet

Ce projet a permis de consolider notre maîtrise des outils de visualisation de données et du développement web interactif.

Points forts

- Interface intuitive
- Comparaison visuelle efficace
- Données pertinentes

Limites

- Nombre restreint de villes
- Données parfois manquantes selon la ville

Améliorations envisagées

- Extension à plus de villes
- Ajout de nouveaux critères (sécurité, transport...)
- Interface multilingue