Projet de web-scraping et de data-visualisation du marché immobilier français

Collaborateurs :
Sufyan Nadat
Jacques Allison
Cédric Manelli

Université Paris Sorbonne

Table des matières

1 Introduction du projet

Le projet présenté s'inscrit dans une démarche d'analyse du marché immobilier français à partir de données issues du web-scraping. Il a été réalisé par trois collaborateurs – Sufyan Nadat, Jacques Allison et Cédric Manelli – et combine des compétences en collecte automatisée de données en web scaping, en traitement statistique, en analyse exploratoire et visualisation interactive.

1.1 Objectif global du projet

L'objectif principal est de constituer une base de données immobilières exploitable, à partir d'annonces publiées sur deux plateformes majeures du secteur : EtreProprio et SeLoger, puis d'en tirer des indicateurs statistiques pertinents permettant d'analyser les prix, les surfaces, les typologies de biens et leur répartition géographique. Le projet vise également à proposer une restitution interactive sous forme d'une application Streamlit, rendant les résultats accessibles à un public semi-technique.

1.2 Problématique métier et technique du projet

Le marché immobilier est caractérisé par une forte hétérogénéité des biens (prix, surfaces, localisation, type de logement) et par une information dispersée sur différentes plateformes. La problématique consiste donc à :

— Collecter des données fiables et suffisamment volumineuses, malgré les contraintes imposées par les sites web (pagination, limites d'affichage, protections anti-scraping).

— Harmoniser des données issues de sources différentes, avec des formats, des champs et des niveaux de complétude variables.

— Nettoyer et filtrer les données afin d'obtenir des indicateurs cohérents et exploitables.

— Restituer l'information de manière lisible, à la fois statistique (graphiques, agrégations) et géographique (cartes interactives).

1.3 Contexte général

Le projet s'inscrit dans un contexte de data analysis appliquée à l'immobilier, domaine à fort enjeu économique. Les données manipulées sont issues d'annonces publiques, structurées puis enrichies (calculs de prix au m^2, regroupements géographiques). L'ensemble de la chaîne – du scraping à la visualisation – est développé intégralement en Python.

2 Description générale du projet

2.1 Nature du projet

Il s'agit d'un projet end-to-end de data engineering et data analysis, comprenant :

— du web-scraping multi-sources,

— du traitement et nettoyage de données,

— de la visualisation statistique et géographique,

— et une application web interactive pour la restitution finale.

2.2 Sources et types de données manipulées

Les données proviennent de deux plateformes distinctes :

— **EtreProprio** : site d'annonces immobilières entre particuliers, exploité via un scraping HTML classique (requests + BeautifulSoup).

— **SeLoger** : plateforme majeure du marché, nécessitant un scraping plus avancé via Selenium, en raison de son contenu dynamique et de ses mécanismes anti-bot.

Les données collectées incluent notamment :

— le prix du bien,

— la surface intérieure (et éventuellement terrain ou extérieure),

— le type de bien (appartement, maison, terrain, commerce),

— le nombre de pièces,

— la ville, le code postal et le département,

— des informations complémentaires comme la classe énergétique ou le caractère « neuf ».

Ces données sont stockées sous forme de fichiers CSV, servant ensuite de base aux traitements analytiques.

2.3 Technologies et outils principaux

Sans dresser une liste exhaustive, les briques techniques majeures sont :

— Python comme langage central,

— Requests / BeautifulSoup pour le scraping statique,

— Selenium (ChromeDriver) pour le scraping dynamique,

— Pandas pour la manipulation et le nettoyage des données,

— Geopy (avec Nominatim) pour la géolocalisation et le géocodage des adresses,

— Folium pour la cartographie interactive,

— Streamlit pour la création de l'application web,

— Plotly pour les graphiques interactifs.

L'architecture globale montre une séparation claire entre collecte, traitement et restitution, ce qui favorise la lisibilité et la maintenabilité du projet.

3 Étapes clés du projet

3.1 Collecte des données

La collecte repose sur deux scripts de scraping distincts, chacun adapté aux contraintes de sa source.

Pour le site EtreProprio, le scraping repose sur l'analyse de la structure HTML, la gestion de la pagination et l'extraction des URLs d'annonces, suivies d'une collecte parallèle des informations clés. Afin de contourner la limite du nombre d'annonces par recherche, nous avons mis en place une stratégie de filtrage. Chaque combinaison de filtres déclenche une nouvelle requête permettant de récupérer un volume maximal d'annonces. Ces filtres incluent notamment le numéro de département, le type de bien, les plages de prix et l'ordre chronologique des annonces. L'objectif était de limiter chaque recherche à un maximum de 600 annonces (20 annonces par page sur 30 pages). Cette approche génère des URLs structurées selon le schéma suivant :

```
etreproprio/{bien_code}.{prix_min}{prix_max}.{dep}{date_order[0]}#list
```

Pour SeLoger, le scraping est plus complexe, c'est pour cette raison que nous allons le détailler. Ce Web scraping utilise la bibliothèque Selenium, un outil qui permet de contrôler un navigateur web de manière programmatique. Concrètement, Selenium ouvre un vrai navigateur, clique sur des boutons, fait défiler les pages et lit le contenu exactement comme le ferait un humain, mais de façon automatisée. C'est différent d'autres méthodes de scraping qui envoient simplement des requêtes HTTP, car ici le navigateur exécute réellement le JavaScript de la page, ce qui est nécessaire pour les sites modernes comme SeLoger détonants des mesures anti-bot.

Le script est capable de fonctionner en parallèle avec plusieurs navigateurs simultanément, ce qui accélère considérablement le processus de collecte de données.

3.1.1 Importations et dépendances

Les bibliothèques les plus importantes sont Selenium pour le contrôle du navigateur, undetected_chromedriver (une version modifiée de Chrome qui aide à éviter la détection

comme robot), concurrent.futures pour exécuter plusieurs navigateurs en parallèle. Le module re (expressions régulières) est aussi essentiel pour analyser et extraire les informations des textes des annonces.

3.1.2 Configuration

Le script définit de nombreux paramètres ajustables. L'URL de base cible une recherche SeLoger pour des maisons et appartements à vendre dans Paris intra-muros.

Ensuite, on utilise des sélecteurs CSS, ce sont les "adresses html" qui permettent de localiser précisément chaque élément sur la page : où se trouve le prix, l'adresse, la surface, etc.

3.1.3 Ce qui distingue ce scraper

La gestion automatique des popups est une fonctionnalité centrale de cette version. Les sites modernes affichent de nombreuses fenêtres surgissantes (consentement cookies, newsletters, publicités) qui bloquent l'accès au contenu. Ce script détecte et ferme automatiquement tous ces obstacles, y compris les popups utilisant le "Shadow DOM", une technique web moderne qui rend les éléments invisibles aux méthodes de scraping classiques. Le script utilise du JavaScript injecté directement dans la page pour contourner cette difficulté.

La simulation du comportement humain et l'évasion des mesures anti-bots sont particulièrement importants. Le défilement de page utilise une courbe d'accélération naturelle (rapide au début, ralentissant à la fin) plutôt qu'un mouvement linéaire robotique. Le script prend des pauses aléatoires toutes les 8 à 15 pages, comme un humain. Il change parfois la taille de sa fenêtre en cours de route.

De plus, des délais sont utilisés pour réguler le rythme d'exécution (chargement, défilement, traitement des annonces). Ces délais sont tous définis comme des plages aléatoires (par exemple entre 2 et 4 secondes) plutôt que des valeurs fixes, ce qui rend le comportement moins détectable comme automatisé.

Enfin, une liste configurable de user agents (identifiants de navigateur) et de résolutions d'écran permet de varier l'apparence de chaque navigateur, comme si différentes personnes utilisaient différents ordinateurs.

3.1.4 Extraction des données

Pour chaque annonce, le script extrait le type de bien, le prix, le prix au mètre carré, la surface, le nombre de pièces et de chambres, l'étage, l'adresse complète avec ville et code postal, la classe énergétique, et le lien vers l'annonce détaillée.

3.1.5 Fonctionnement parallèle

Enfin, le script peut ouvrir plusieurs navigateurs Chrome simultanément (par défaut 3, maximum 10). Les pages à scraper sont réparties équitablement entre ces navigateurs qui travaillent indépendamment.

3.2 Contraintes liées à l'extraction des données

Les principales contraintes proviennent des sites sources :

— limitations de pagination (notamment sur EtreProprio),

— blocage des bots d'extraction sur le site SeLoger,

— structures HTML changeantes,

— données manquantes ou hétérogènes selon les annonces.

Pour SeLoger, la présence de protections anti-scraping impose :

— l'utilisation de Selenium,

— la gestion de pop-ups et de contenus dynamiques,

— une simulation de comportement humain (scroll, pauses, variations d'agent utilisateur).

Ces contraintes ont un impact direct sur la performance et la complexité du code.

3.3 Choix techniques majeurs

Le choix de séparer les scrapers par source est pertinent, car il permet d'adapter finement la logique à chaque site.

L'utilisation du multithreading pour la collecte améliore significativement les temps d'exécution, tout en restant contrôlée pour éviter les blocages.

3.4 Prétraitement et nettoyage

Une fois les fichiers CSV générés, un notebook dédié se charge :

— de convertir les variables en formats numériques cohérents,

— d'exclure les annonces incomplètes ou aberrantes,

— d'assurer la cohérence entre les noms des variables, le type de variable, le prix, le type de surfaces et la localisation,

— de créer une variable prix au m^2.

Cette étape est centrale pour garantir la qualité des analyses ultérieures.

3.5 Géocodage et spatialisation des données

Pour spatialiser les annonces, nous avons utilisé le nom du département renseigné pour chaque annonce dans le fichier CSV afin d'associer des coordonnées géographiques à l'échelle départementale. La même approche a été appliquée aux arrondissements de Paris.

Méthodologie : Les coordonnées géographiques ont été obtenues à l'aide de l'API Nominatim (via la bibliothèque geopy). Cette étape a permis d'extraire les latitudes et longitudes associées à chaque département. La même procédure a ensuite été appliquée aux arrondissements de Paris.

Résultat : Deux fichiers GeoJSON personnalisés ont été générés : le premier contient les coordonnées géographiques (latitude et longitude) de chaque département de la France, tandis que le second rassemble celles de chaque arrondissement de Paris.

3.6 Création de cartes (Folium)

À l'aide de la bibliothèque Folium, nous avons généré deux cartes interactives reposant sur une représentation en « bulles », à partir des fichiers GeoJSON précédemment construits :

— **Échelle Nationale** : Une carte de la France sur laquelle chaque département est représenté par un point géographique. La taille des bulles est proportionnelle au volume d'annonces recensées. Un clic sur chaque point permet d'afficher le prix moyen au mètre carré associé au département, tandis qu'un code couleur, allant du vert au rouge, traduit visuellement les niveaux de prix, des plus faibles aux plus élevés.

— **Échelle Parisienne** : Une visualisation plus fine à l'échelle des arrondissements de Paris, fondée sur les mêmes principes de représentation.

Ces cartes transforment les données brutes en un outil d'analyse spatiale interactif, rendant immédiatement perceptibles la répartition de l'offre immobilière ainsi que les disparités de prix entre territoires.

3.7 Analyse et modélisation

L'analyse reste volontairement exploratoire et descriptive. Elle repose sur des agrégations (moyennes, distributions), des comparaisons entre villes ou départements et l'étude de relations simples, comme celle entre surface et prix.

4 Résultats obtenus et présentation Streamlit

4.1 Volume et structure des données

Les données finales regroupent plus de 617 000 annonces, selon les zones géographiques analysées. Chaque ligne correspond à un bien immobilier, avec des variables numériques et catégorielles cohérentes tels que le type de bien, la surface du terrain (pour les terrains nus), la surface intérieure, la surface extérieure, le nombre de pièces, le prix, le prix/m^2, certaines adresses ciblées sur Paris, la ville, le code postal, le département et le code département.

Les données sont présentées dans une application Streamlit, une application de présentation et d'exploration du marché immobilier en France, centrée sur des annonces stockées dans un fichier CSV. L'objectif est de proposer une interface qui affiche des indicateurs clés, des graphiques interactifs Plotly et deux cartes HTML (France + Paris). L'utilisateur peut filtrer certaines vues (département, ville, arrondissement) pour obtenir une lecture plus fine des prix et de leur distribution, et repérer des écarts territoriaux.

4.2 Structure de l'application

L'application se découpe en quatre grandes sections :

— **Présentation du projet** : rappel du projet, des sites utilisés pour le scraping, des méthodes et du traitement des résultats.

— **Statistiques** : une zone KPI accompagnée de trois onglets de graphiques.

— **Cartographie des prix** : deux onglets avec cartes interactives.

— **Pied de page** : rappel « Données immobilières • Visualisation interactive ».

Le chargement des données est conditionné à la présence d'un fichier CSV. En l'absence de celui-ci, l'application affiche un message d'avertissement et désactive la partie statistique.

4.3 Indicateurs clés de performances (KPI)

Le dashboard calcule et affiche les indicateurs suivants :

— Nombre d'annonces (taille du dataframe),

— Prix moyen,

— Prix moyen au m^2,

— Surface intérieure moyenne.

Ces valeurs donnent une vision macro du marché, mais restent sensibles aux valeurs extrêmes, car elles reposent sur des moyennes simples, sans recours à la médiane.

4.4 Graphiques interactifs

4.4.1 Prix moyen par type de bien

Ce graphique prend la forme de barres horizontales. Il est filtrable par département et par type de bien, et affiche à la fois le prix moyen et le nombre d'annonces. Il permet de comparer la hiérarchie des types de biens (par exemple appartement versus maison) et d'observer les écarts de prix selon les départements.

4.4.2 Distribution du prix au m² par ville et par arrondissement parisien

L'utilisateur peut sélectionner l'option « Toutes les villes », une ville du top 30 (hors Paris) ou « Paris ». Pour Paris, il est possible de descendre au niveau de l'arrondissement. La courbe représente une densité transformée en pourcentage, permettant de visualiser les zones de concentration des annonces en euros par mètre carré (forme unimodale ou bimodale, dispersion, présence d'une queue haute).

4.4.3 Prix moyen au m² par département

Ce graphique présente les 15 départements les plus chers selon le prix moyen au m^2, sous forme de barres horizontales colorées. Le code couleur est lié au niveau de prix, et le survol permet d'afficher le volume d'annonces associé.

4.5 Cartes interactives

Deux cartes interactives sont proposées :

— une carte de la France par département,

— une carte de Paris par arrondissement.

La légende indique que la couleur représente le prix moyen au m^2, tandis que la taille des cercles correspond au nombre d'annonces. Les cartes sont externes au script principal et reposent sur des fichiers HTML déjà générés.

4.6 Limites et hypothèses

L'analyse est centrée sur des moyennes, sensibles aux valeurs aberrantes que nous avons tenté d'exclure au maximum. Elle n'intègre pas de médianes, de quartiles ou de segmentations fines (neuf/ancien, localisation précise, etc.). Avec davantage de détails sur le jeu de données, il aurait été possible d'évaluer plus finement la qualité des biens (neuf, nombre de fenêtres par pièce, état du bien, classe énergétique, etc.).

La qualité des résultats dépend fortement des données collectées et des traitements appliqués. Ce dashboard Streamlit constitue avant tout un outil de valorisation du projet de data analysis immobilière. Il combine des KPI, des comparaisons par typologie, des

distributions de prix au m^2 et une cartographie France/Paris. Pour une conclusion data plus robuste, une évolution logique consisterait à intégrer des médianes et des quantiles, des contrôles d'outliers plus stricts et des analyses explicatives supplémentaires, telles que la corrélation entre prix et surface ou la segmentation par nombre de pièces et par département.

5 Limites du projet

5.1 Couverture des données

Le projet se limite à deux plateformes et ne prétend pas couvrir l'ensemble du marché immobilier français. Certaines typologies de biens ou zones géographiques peuvent être sous-représentées.

5.2 Qualité et fraîcheur des données

Les données sont figées à un instant donné. Elles ne prennent pas en compte l'évolution temporelle des prix ni la durée de publication des annonces. Nous n'avions aucune date d'apparition des biens sur les sites.

5.3 Hypothèses simplificatrices

Le calcul du prix au m^2 repose sur des surfaces déclarées, parfois approximatives. Les effets de qualité du bien, de l'état général ou du standing ne sont pas intégrés (comme les surfaces extérieures par exemple).

5.4 Aspects non traités

Le projet ne propose pas de modélisation prédictive ni d'analyse temporelle avancée. Il reste volontairement centré sur une analyse descriptive et exploratoire.

6 Conclusion

Ce projet de web-scraping et d'analyse immobilière constitue une réalisation complète et cohérente, couvrant l'ensemble de la chaîne de valeur de la donnée : de la collecte brute à la restitution interactive. Il met en évidence une bonne maîtrise des outils Python, des problématiques de scraping réel et des méthodes de visualisation modernes.

L'intérêt principal du projet réside dans sa dimension intégrée et dans la qualité de la restitution finale, notamment via Streamlit et la cartographie Folium. Il offre une

base solide pour des évolutions futures, telles que l'ajout d'une dimension temporelle, l'intégration d'autres sources de données ou le développement de modèles prédictifs.

En synthèse, il s'agit d'un projet techniquement abouti, pertinent sur le plan métier, et représentatif d'une démarche professionnelle en data analysis appliquée à l'immobilier.