Lina Ibouchichene
Alexandre Mégard

Analyse du marché immobilier à partir de données web scraping avec Python

Année universitaire 2025–2026

Table des matières

I) Introduction

« Après plus de trois ans de recul, le marché immobilier ancien montre enfin des signes tangibles de reprise. Les ventes repartent à la hausse et les prix se stabilisent, soutenus par des taux d'intérêt désormais stables autour de 3 % ». (Journal de l'Agence, 11 Novembre 2025). À la suite de la période post-COVID, l'économie a été fragilisée et le secteur immobilier en a subi les conséquences, rendant l'analyse des marchés immobiliers plus critique. Le marché immobilier occupe une place centrale dans l'économie française et connaît depuis plusieurs années de fortes variations de prix selon les territoires. Dans ce contexte, il est impératif de disposer d'une analyse fine des tendances de prix.

Les plateformes d'annonces en ligne, telles que Leboncoin, SeLoger ou Bien'ici, regroupent une grande quantité de données brutes reflétant l'offre actuelle du marché immobilier. L'exploitation de ces données par des méthodes d'automatisation permet d'identifier des tendances locales que les statistiques traditionnelles ne prennent pas forcément en compte. Ainsi, ce projet s'articule autour de la problématique suivante :

Comment le prix au mètre carré varie-t-il en fonction de la localisation, de la surface et du type de bien immobilier en France ?

Pour répondre à cette question, un site immobilier est scrappé (*ÊtreProprio*) afin de constituer une base de données contenant le prix, la surface, la localisation, le nombre de pièces et le type de bien. Les données collectées sont ensuite nettoyées et analysées à l'aide d'outils statistiques et interpréter avec des outils de visualisations. Enfin, un tableau de bord interactif est développé via Streamlit afin de faciliter l'exploration de ces résultats.

II) Étude de l'art

Le web scraping occupe aujourd'hui une place essentielle dans la collecte automatisée de données issues du Web. C'est une technique permettant de récupérer des informations présentes sur des pages en ligne en imitant le comportement d'un utilisateur humain. Dans de nombreux secteurs, ce procédé est devenu une méthode incontournable. En effet, les plateformes d'annonces immobilières centralisent un volume important de données, mais celles-ci ne sont pas homogènes, ne sont pas toujours accessibles via une API et ne sont consultables qu'à travers des pages HTML individuelles. Le web scraping permet alors d'automatiser l'extraction d'informations telles que les prix, les surfaces, les localisations ou les types de biens, afin de constituer une base de données structurée et exploitable.

Le principe général du scraping repose sur plusieurs étapes successives : l'envoi de requêtes HTTP vers des pages web, l'analyse de leur contenu HTML afin d'identifier les balises pertinentes, puis l'extraction et le stockage des données dans un format adapté à l'analyse, généralement sous forme de tableaux ou de fichiers CSV. C'est cette démarche qui a été adoptée dans le cadre de ce projet.
La mise en œuvre du scraping repose sur l'utilisation de bibliothèques Python adaptées. La librairie requests permet d'envoyer des requêtes HTTP et d'accéder aux pages listant les annonces

immobilières, tandis que BeautifulSoup est utilisée pour analyser la structure des pages HTML et extraire les informations pertinentes. La bibliothèque pandas permet quant à elle de structurer, nettoyer et sauvegarder les données extraites sous forme de fichiers CSV. Une approche en deux étapes a été retenue pour ce projet : tout d'abord une première phase de collecte des annonces sur la page principal puis une deuxième phase de scraping à partir des pages individuelles de chaque bien.
Le scraping impose également le respect de certaines bonnes pratiques afin d'éviter les systèmes de détection automatique. L'ajout d'un User-Agent et l'intégration de temps d'attente entre les requêtes permettent de reproduire un comportement humain et de limiter les risques de blocage.

Enfin, plusieurs études et baromètres immobiliers publiés par des acteurs spécialisés tels que MeilleursAgents, SeLoger, PAP ou les Notaires de France analysent régulièrement l'évolution des prix immobiliers. Le web scraping constitue alors une alternative pertinente pour constituer un jeu de données personnalisé et actualisé à partir de données accessibles en ligne. Des travaux académiques récents montrent également l'utilisation du scraping pour analyser le marché immobilier, notamment durant la crise du Covid-19 ou dans le secteur de la location, confirmant ainsi la fiabilité et l'intérêt de cette méthode.

En effectuant quelques recherches nous avons trouvé des études de marchés existantes sur ce domaine, comme cette étude sur les prix de l'immobilier pendant la crise du Covid-19 :

Web-scraping housing prices in real-time: The Covid-19 crisis in the UK

Ou encore cette étude qui porte sur une partie du marché immobilier, plus précisément de la location qui ont utilisé le web scraping pour leur étude :

Can big data increase our knowledge of local rental markets? A dataset on the rental sector in France

Ainsi, l'état de l'art montre que le web scraping, associé aux outils Python tels que requests, BeautifulSoup et pandas, constitue aujourd'hui une méthode fiable, flexible et largement adoptée pour collecter des informations en grande quantité. Il s'agit d'un outil efficace pour analyser des marchés en ligne, et tout particulièrement le marché immobilier, où la diversité des plateformes et la fréquence de mise à jour des annonces rendent le marché immobilier riche et complexe.

III) Méthodologie

1) Description du site cible

Le projet repose sur la collecte de données issues du site d'annonces immobilières *etreproprio.com*. Ce site contient un volume important d'annonces provenant de professionnels comme de particuliers et présente des informations standardisées sur chaque bien : titre, description, localisation, surface, prix, nombre de pièces et type de logement. Il est exclusivement dédié à la vente de biens immobiliers. Le choix de se focaliser uniquement sur la vente, et non sur la location, se justifie par la nécessité de disposer d'un indicateur de valeur immobilière plus stable et directement comparable pour analyser la variation du prix au mètre carré. En effet, le prix de vente représente la valeur réelle du bien et constitue un indicateur moins volatil que le loyer, souvent influencé par des facteurs réglementaires ou subjectifs. De plus, sa structure de page est plus simple que celle de plateformes comme SeLoger ou Leboncoin, qui reposent davantage sur des scripts dynamiques et des injections JavaScript, compliquant le scraping. Dans ce cas, l'utilisation des bibliothèques *requests* et *BeautifulSoup* s'est avérée suffisante, sans recourir à des outils plus lourds comme Selenium. De plus, le site regroupe un nombre significatif d'annonces réparties sur l'ensemble du territoire français, ce qui a permis de constituer un jeu de données varié couvrant une quarantaine de villes.

Pour en revenir au site, son fonctionnement repose sur un système de pages de résultats organisées par ville, chaque commune étant identifiée par un code propre au site. Ces pages affichent des listes d'annonces sous forme de blocs répétitifs, 20 par pages, ce qui en fait une structure adaptée au scraping, car les éléments HTML qui composent une annonce apparaissent de manière régulière et similaire partout donc facilement accessibles via des sélecteurs CSS.

Chaque ville est accessible via une URL structurée autour d'un code interne (par exemple "thflcpo.lc69266" qui regroupe les appartements, maisons, commerces de la ville de Villeurbanne, si c'étaient seulement les appartements par exemple le code aurait été légèrement différent : tf.lc69266). Il existe également une pagination qui permet de naviguer entre plusieurs pages de résultats pour une même ville, chacune contenant un nombre limité d'annonces, 20 par page. Le site offre ainsi une architecture plutôt simple à analyser : un identifiant de ville, un numéro de page et une liste d'annonces présentée sous une structure HTML stable. Cette organisation a facilité l'extraction d'informations, mais a aussi nécessité une étude précise des classes CSS utilisées dans les balises du site, notamment celles contenant les titres (".ep-title"), les prix (".ep-price"), les surfaces (".ep-area"), les villes (".ep-city") et les liens vers les pages individuelles ("a" parent du bloc). Il fallait aller chercher les balises correspondantes manuellement, par exemple pour la ville il y avait plusieurs balises qui affichaient le nom de la ville, il fallait aller chercher la plus intéressante.

2) Architecture technique du projet

L'architecture du pipeline de scraping repose sur une séparation claire entre deux étapes complémentaires : une première phase d'extraction large des annonces présentes dans les pages de résultats de recherche, puis une seconde phase de scraping approfondie des informations contenues dans les pages détaillées de chaque annonce.

Dans la première partie, le script parcourt successivement 41 villes françaises représentées par une liste de codes, que nous avons au préalable récupéré une par une. Pour chaque ville, il parcourt six pages de résultats, ce qui permet d'obtenir un échantillon suffisamment large tout en maintenant un temps d'exécution assez court. Cette phase utilise la fonction *get_page*, qui envoie une requête HTTP vers l'url correspondant à une combinaison "{city_code}r0.odd.g{page_number}", qui conserve le code de la ville et la page. Ensuite on convertit le contenu en un objet BeautifulSoup.

```
def get_page(city_code, page_number):
    url = f"https://www.etreproprio.com/annonces/{city_code}-r0.odd.g{page_number}"
    response = requests.get(url, headers=HEADERS, timeout=10)
    return BeautifulSoup(response.content, "lxml")
```

Les annonces, reconnues grâce à la classe HTML ".card-cla-search", qui nous permet de cibler plus précisément cet élément HTLM : « », sont ensuite extraites une par une via la fonction *extract_info*. Cette fonction récupère les informations immédiatement visibles dans la liste : titre, ville, prix, surface et lien (url). Les données ainsi collectées sont stockées directement dans une liste Python que l'on a appelé data puis exportées sous forme d'un premier fichier CSV nommé *ANNONCES_RAW.csv* que l'on initialise juste après notre boucle de scraping. Concernant son architecture d'abord on initialise un hearder agent puis la première fonction get_page avec la requête http puis la fonction extract_info qui récupères les informations dans les balises, puis une liste des villes avec leur code et enfin la boucle de scraping.

La deuxième phase se concentre exclusivement sur les pages individuelles de chaque annonce, accessibles grâce aux URLs collectées au préalable. Cette étape a pour objectif d'enrichir les données en récupérant des éléments qui n'apparaissent pas dans les pages de résultats, comme le code postal, le type de bien ou le nombre de pièces. Le script lit donc le fichier *ANNONCES_RAW.csv*, extrait la liste des URLs uniques, puis parcourt chacune d'elles via la fonction *scrape_detail_page*.

```
HEADERS = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0 Safari/537.36",
    "Accept-Language": "fr-FR,fr;q=0.9"
}

INPUT_FILE = "ANNONCES_RAW.csv"
OUTPUT_FILE = "ANNONCES_DETAILLEES.csv"

def scrape_detail_page(url):
    """Scrape une page d'annonce EtreProprio et retourne un dictionnaire d'infos."""

    try:
        r = requests.get(url, headers=HEADERS, timeout=10)
        soup = BeautifulSoup(r.content, "lxml")

    except:
        print(f" Erreur de connexion pour : {url}")
        return None
```

Cette fonction analyse le contenu HTML de la page du bien, et repère des éléments plus complexes comme la localisation complète (code postal + ville) , qui nécessite un traitement du texte brut pour séparer le nom de la ville et le code postal (qui sont de base dans le même élément HTML : « Villeurbanne 69100 »). Mais aussi pour le type de bien que l'on est parti chercher dans le fil d'ariane en haut de la page, on va chercher à récupérer seulement le 2ème terme de la ligne, en l'indiquant comme ceci « li:nth-child(2) ». Ensuite on crée notre boucle de scraping, cette fois ci en y incluant plus de pauses que l'on détaillera juste en dessous. Une fois les données extraites, elles sont ajoutées ligne par ligne dans un second fichier CSV, intitulé *ANNONCES_DETAILLEES.csv*, qui représente la version finale et enrichie du dataset, avant qu'il soit nettoyé dans un second temps.

Quelques exemples :

```
# --- Ville et Code Postal ----
  loc_tag = soup.select_one(".ep-loc")

  ville = None
  code_postal = None

  if loc_tag:

      texte = loc_tag.get_text(" ", strip=True)
      texte = texte.replace("—", "").strip()
      elements = texte.split()
      code_postal = elements[-1]
      ville = " ".join(elements[:-1])

  # ---- Type du bien ----
  type_tag = soup.select_one("div.ep-breadcrumb-cla-dir ol li:nth-child(2)")
  type_bien = type_tag.get_text(strip=True) if type_tag else None
```

Cette architecture duale présente plusieurs avantages. Elle permet d'abord de réduire le risque d'erreur, car le système ne dépend pas d'une seule grosse boucle qui ferait tout en une fois. En effet la deuxième partie du scraping a pris plus de 3h pour charger toutes les annonces, il ne faut pas avoir de problèmes de connexion ou d'interruption car on peut perdre le chargement de plusieurs annonces. Ce qui facilite également la reprise du processus en cas d'interruption, puisqu'il suffit de relancer la seconde partie à partir du fichier des URLs. Enfin, elle améliore la lisibilité du code, chaque étape ayant une responsabilité clairement définie : extraction générale, puis extraction détaillée.

3) Stratégie de scraping : pagination, exploration et gestion anti-bot

La mise en œuvre du scraping a nécessité une stratégie rigoureuse pour s'adapter à la structure du site et éviter tout blocage lié aux mécanismes anti-bot. La première étape a consisté à analyser la manière dont le site organise sa pagination. Sur EtreProprio, chaque page de résultats suit un schéma d'URL fixe, dans lequel seul le numéro de page varie. Cette régularité a permis de construire des urls grâce à la fonction *get_page*, qui combine un code ville et un entier correspondant au numéro de page. Si une page ne contient plus aucune annonce, le scraping de cette ville s'arrête automatiquement et passe à la ville suivante. Cette stratégie permet d'éviter les requêtes inutiles et d'adapter la stratégie de scraping en fonction du volume réel d'annonces.

La gestion des protections anti-bot constitue un autre élément essentiel dans la conception de notre scraping. Les sites peuvent identifier des comportements suspects, comme un nombre trop élevé de requêtes en peu de temps ou l'utilisation d'un User-Agent inhabituel. Pour simuler un comportement humain, nous avons inclu User-Agent identique à celui d'un navigateur Chrome. De plus, le script introduit des délais entre les requêtes grâce à la fonction *time.sleep*, en utilisant des durées aléatoires générées par *random.uniform* entre 1.2 et 2.8 secondes, puis dans la seconde partie, des pauses de 10 secondes toutes les 20 pages qui permettent également de ralentir le robot et de diminuer davantage la probabilité de déclencher les protections du serveur. Cette gestion aléatoire du temps d'attente évite que les requêtes soient envoyées à un rythme trop régulier, ce qui serait facilement détectable comme une activité robotique.

La stratégie d'extraction elle-même se décompose en deux niveaux. Au premier niveau, le script se contente de parcourir les pages généralistes pour collecter les informations immédiatement visibles. Ce sont ces pages qui contiennent un volume élevé d'annonces, ce qui permet de constituer rapidement un dataset initial. Au second niveau, chaque annonce est explorée plus en profondeur. Cette approche hiérarchique présente un intérêt majeur : elle évite de télécharger inutilement des centaines de pages détaillées si les informations de base ne sont pas pertinentes ou si l'annonce ne contient pas de liens valides. Le scraping devient ainsi plus rapide, plus propre, et mieux structuré.

4) Processus de nettoyage, structuration et enrichissement des données

Une fois les données collectées à travers les deux phases de scraping, une étape essentielle persiste, celle qui consiste à nettoyer et structurer le dataset afin de le rendre exploitable pour l'analyse statistique et géographique que nous allons faire juste après. Les données issues du web scraping sont en effet hétérogènes, parfois incomplètes, et fréquemment mélangées à des caractères non numériques, ce qui rend indispensable la phase de nettoyage approfondi. Le script de nettoyage que nous avons créé permet de corriger les incohérences, extraire les éléments numériques pertinents et enrichir le dataset avec des informations de localisation géographique et de prix au m2.

La première opération effectuée consiste à éliminer les doublons. Lors du scraping, certaines annonces peuvent apparaître plusieurs fois, notamment lorsqu'une même annonce figure sur plusieurs pages. La suppression des doublons assure ainsi que chaque bien immobilier ne figure qu'une seule fois dans la base, ce qui évite de biaiser les statistiques finales.

Ensuite on nettoie les colonnes contenant des valeurs numériques. Les prix affichés sur les annonces sont généralement accompagnés du symbole euro, d'espaces ou de mentions particulières. Le script supprime donc successivement le symbole "€", les espaces et les chaînes non pertinentes avant de convertir la colonne en type numérique. La même démarche est appliquée à la colonne "surface".

À partir de ces données nettoyées, il devient possible de calculer le prix au mètre carré, un indicateur central de l'analyse immobilière. Ce calcul repose sur la division du prix total par la surface du bien. Ce nouvel indicateur enrichit le dataset et permet une comparaison plus pertinente entre des biens situés dans des zones différentes ou appartenant à des catégories variées.

L'une des étapes les plus importantes du processus de nettoyage concerne la géolocalisation des villes. Pour les besoins de la visualisation cartographique et de l'analyse spatiale, il est indispensable de disposer de coordonnées GPS (latitude et longitude) associées à chaque ville. À cette fin, le script utilise la bibliothèque geopy et le service Nominatim d'OpenStreetMap. Ces valeurs sont ensuite ajoutées sous forme de deux nouvelles colonnes, "latitude" et "longitude".

Enfin, le dataset nettoyé est sauvegardé dans un nouveau fichier intitulé ANNONCES_CLEAN.csv. Ce fichier représente la base finale, prête pour l'analyse statistique, la production de graphiques, de cartes interactives et la construction du tableau de bord.

IV) Résultats et analyses

Statistiques clés du jeu de données

Nombre total d'annonces : 4920
Nombre d'appartements : 3441
Nombre de villes : 42

===== NOMBRE D'APPARTEMENTS PAR VILLE =====

Strasbourg	109
Saint-Etienne	104
Grenoble	103
Villeurbanne	103
Clermont-Ferrand	101
Metz	99
Vincennes	98
Rouen	98
Dijon	98
Annecy	97
Nice	95
Rennes	95
Paris	93
Nancy	92
Toulon	92
Reims	92
Lille	91
Mulhouse	91
La Rochelle	59
Poitiers	48

Nombre total d'annonces : 4920
Nombre d'appartements : 3441
Nombre de villes : 42

Pau	90
Toulouse	89
Brest	88
Nantes	87
Tours	85
Montpellier	81
Montreuil	81
Perpignan	80
Bordeaux	78
Limoges	77
Le Havre	76
Orleans	75
Caen	75
Nimes	74
Angers	71
Aix-En-Provence	71
Avignon	69
Saint-Denis	68
Chartres	65
Le Mans	59
Amiens	43

===== PRIX MOYEN AU M2 PAR VILLE =====

Paris	10769.036247
Vincennes	8926.317669
Nice	6718.783647
Annecy	6657.331654
Aix-En-Provence	6347.742096
Montreuil	5982.716379
La Rochelle	5566.281292
Bordeaux	4500.377830
Rennes	4408.316334
Montpellier	4217.475702
Lille	4091.147043
Ceret	3953.846154
Strasbourg	3829.812642
Nantes	3799.838516
Villeurbanne	3793.200049
Toulouse	3766.983031
Angers	3748.967605
Toulon	3558.802854
Caen	3357.477241
Tours	3254.671947
Saint-Denis	3107.423909

===== STATISTIQUES PRIX AU M² =====

Prix moyen au m² : 3803 €
Prix médian au m² : 3204 €

====== CORRÉLATION ======

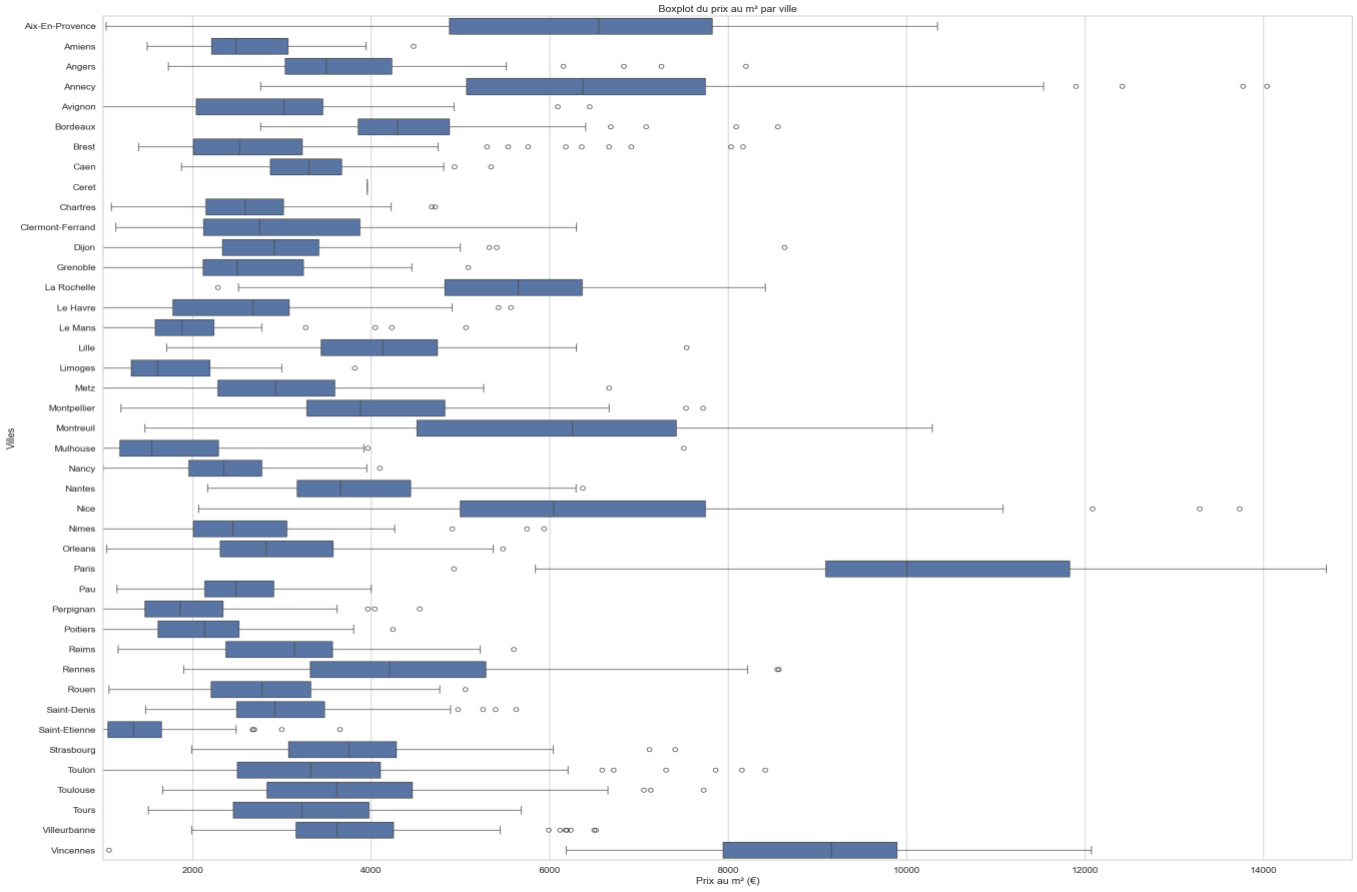
Corrélation surface/prix : 0.306

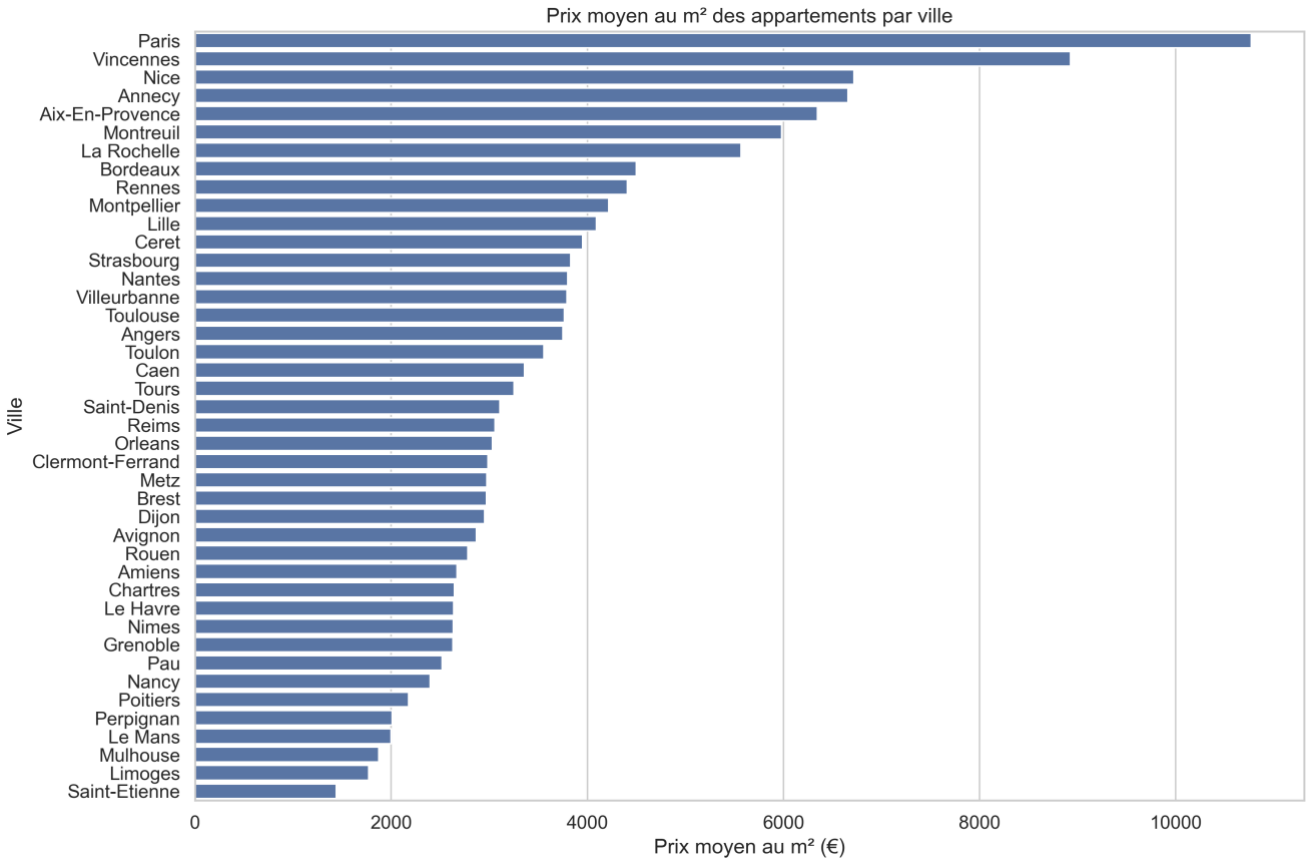
Reims	3057.778415
Orleans	3033.887012
Clermont-Ferrand	2986.137911
Metz	2973.746693
Brest	2973.588774
Dijon	2952.028421
Avignon	2868.562972
Rouen	2780.873053
Amiens	2671.894317
Chartres	2646.118958
Le Havre	2638.218249
Nimes	2633.010957
Grenoble	2630.943275
Pau	2520.392180
Nancy	2397.226688
Poitiers	2178.873554
Perpignan	2008.965343
Le Mans	1998.857248
Mulhouse	1873.734779
Limoges	1769.884638
Saint-Etienne	1441.062409

Pour cette partie d'analyse on a décidé dans un premier temps de se concentrer exclusivement sur les appartements et non sur les maisons et commerces, en effet il y a plus d'appartements dans le csv donc l'analyse sera plus complète car chaque ville a au minimum 43 appartements pour la ville avec le moins et celle avec le plus, 109 appartements, donc pour l'analyse c'est mieux, de plus le prix des maisons est plus ou moins très différents selon la taille et le m2.

Nous avons effectué plusieurs calculs statistiques, notamment le **prix moyen au m² (3 803 €)** et le **prix médian au m² (3 204 €).** L'écart significatif entre ces deux indicateurs, supérieur à **600 €**, indique qu'une minorité de biens très chers, situés dans des zones à forte valeur immobilière, tire la moyenne vers le haut.

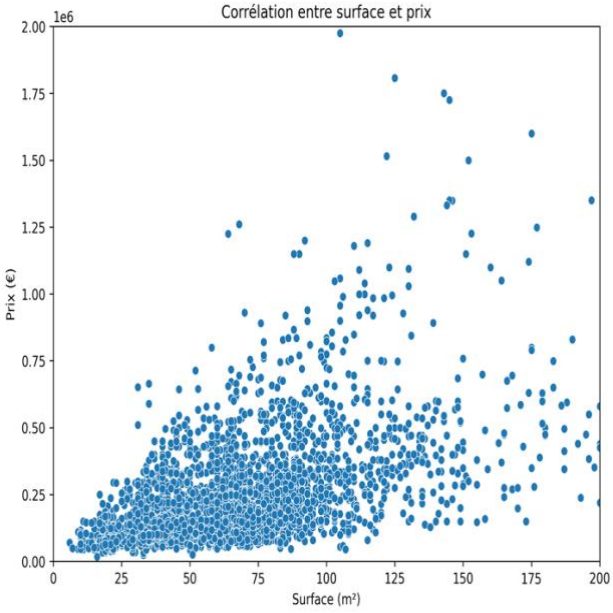
Cette observation se confirme avec le boxplot et l'histogramme du prix au m² par ville (cf. figures ci-dessous). Paris apparaît nettement comme la ville la plus chère en termes de prix au m². À l'inverse, des villes comme Saint-Étienne, Mulhouse, Limoges ou Poitiers présentent des médianes beaucoup plus faibles et des distributions plus resserrées.

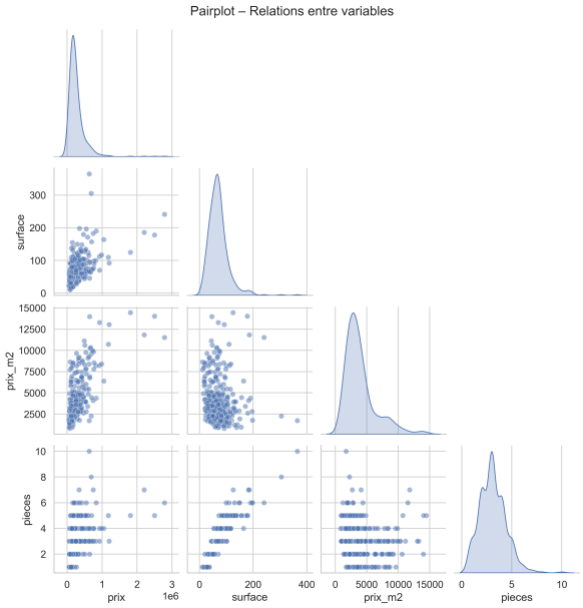



Prix moyen au m² des appartements par ville
Paris
Vincennes
Nice
Annecy
Aix-En-Provence
Montreuil
La Rochelle
Bordeaux
Rennes
Montpellier
Lille
Ceret
Strasbourg
Nantes
Villeurbanne
Toulouse
Angers
Toulon
Caen
Tours
Saint-Denis
Reims
Orleans
Clermont-Ferrand
Metz
Brest
Dijon
Avignon
Rouen
Amiens
Chartres
Le Havre
Nimes
Grenoble
Pau
Nancy
Poitiers
Perpignan
Le Mans
Mulhouse
Limoges
Saint-Etienne
Ville
0
2000
4000
6000
8000
10000
Prix moyen au m² (€)

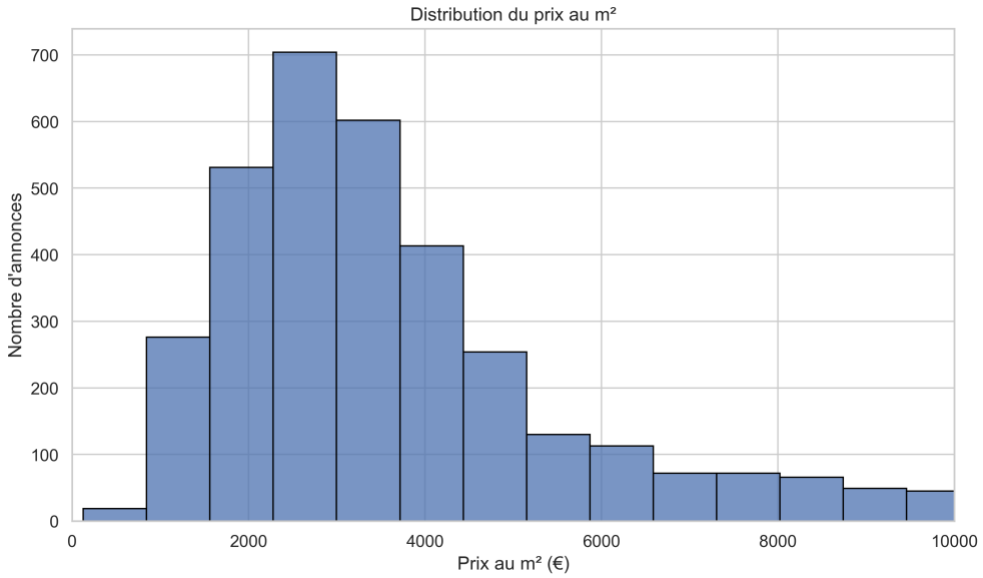
L’étude de la corrélation entre la surface et le prix des appartements met en évidence une relation positive mais relativement modérée, avec un **coefficient de corrélation de 0,306**. Ce résultat indique qu’une augmentation de la surface tend globalement à s’accompagner d’une hausse du prix du bien, mais que cette relation reste loin d’être forte.

D’autres facteurs, tels que la localisation, l’état du logement ou encore les caractéristiques du quartier, expliquent une part importante de la variance des prix. Cette interprétation est confirmée par le nuage de points (cf Corrélation entre surface et prix) qui, bien qu’orienté à la hausse, présente une dispersion importante. Il existe ainsi de nombreux cas où des logements de surface similaires se vendent à des prix très différents, soulignant le rôle déterminant du contexte géographique et des variables subjectives. Cette corrélation modérée entre prix et surface est aussi illustrée graphiquement dans le pairplo**t** ci-dessous. On remarque que si la tendance globale est à la hausse, la dispersion des points est très marquée. Le pairplot permet également de constater que d'autres variables, comme la ville ou le nombre de pièces, se croisent avec la surface pour expliquer la formation du prix final.



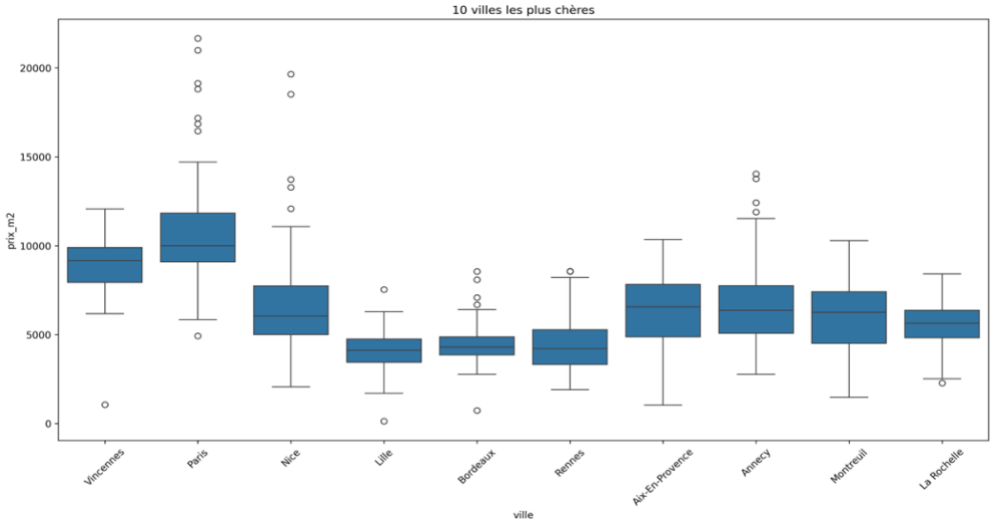


L’histogramme du prix au m² révèle une distribution concentrée entre 2 000 et 4 000 €, avec une traîne vers la droite qui confirme la présence de valeurs extrêmes. Cette asymétrie traduit une segmentation du marché entre les villes dites « standards » et les villes très chères, qui ne représentent qu’une fraction du nombre total d’annonces mais influencent fortement les statistiques globales (cf Distribution du prix au m2).

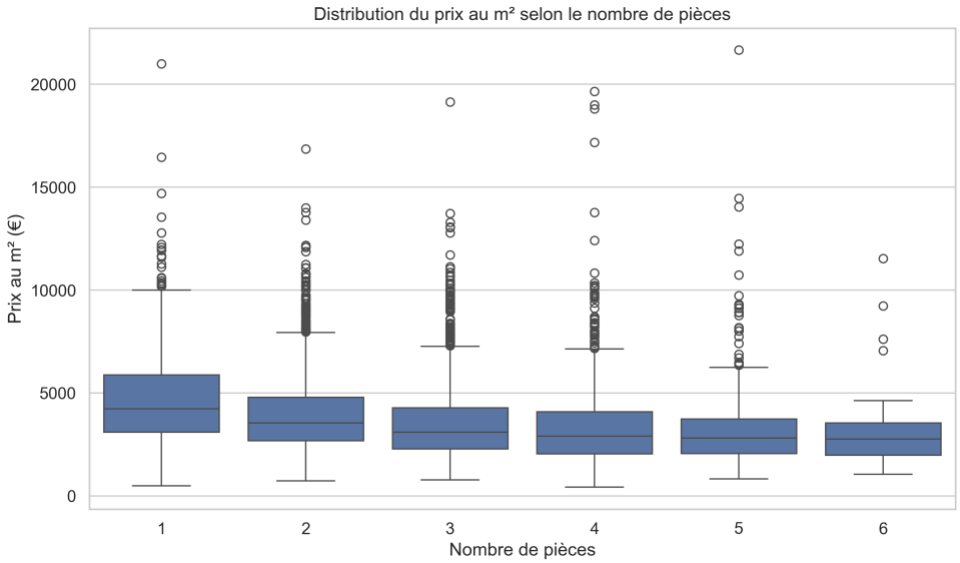


L’analyse détaillée des dix villes les plus chères révèle des disparités majeures au sein même du segment haut de gamme. **Paris** et **Vincennes** se détachent nettement avec des médianes proches ou supérieures à **10 000 €/m².** On observe également une présence importante de "valeurs aberrantes" pour **Paris, Nice et Annecy**, ce qui témoigne de l'existence de biens en particulier dont les prix s'envolent bien au-delà du marché standard.

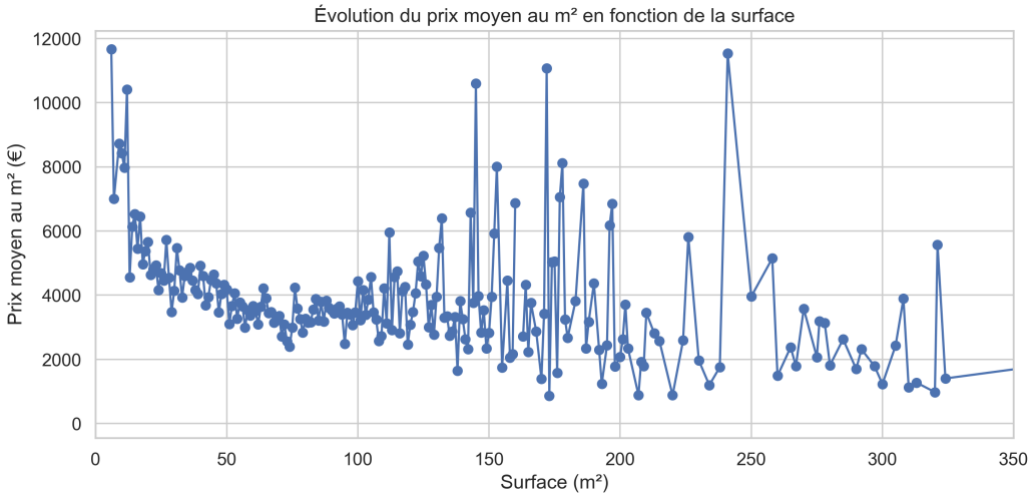
À l'inverse, des villes comme **Lille** ou **Bordeaux**, bien que dans le top 10, présentent des distributions plus resserrées, indiquant un marché haut de gamme plus homogène.



L’analyse du prix au m² selon le nombre de pièces met en évidence une tendance décroissante. Les logements de petite taille, en particulier les studios et deux pièces, présentent un prix au m² plus élevé que les logements plus grands. Cette différence s’explique notamment par une forte demande pour les petites surfaces, en particulier dans les zones urbaines denses, par exemple, c’est ce que montre le boxplot ci-dessous.



L’évolution du prix moyen au m² en fonction de la surface révèle une relation non linéaire. Les petites surfaces affichent des prix au m² plus élevés, tandis que ceux-ci tendent à diminuer à mesure que la surface augmente. Ce phénomène s’explique par un effet de volume, les acheteurs bénéficiant de prix unitaires plus faibles pour les logements de grande taille. Cette dynamique confirme que la surface influence non seulement le prix total, mais également la valeur au mètre carré (cf. Évolution du prix moyen au m² en fonction de la surface).



L’analyse descriptive met en évidence une forte hétérogénéité du marché immobilier étudié. Les prix au mètre carré sont majoritairement concentrés entre 2 000 et 4 000 €/m², mais la présence de certaines villes très chères génère une traîne vers la droite et tire la moyenne vers le haut. Les boxplots confirment ces écarts territoriaux, opposant des marchés relativement homogènes et accessibles à des villes où les prix et la dispersion sont nettement plus élevés.

La surface et le nombre de pièces influencent les prix, mais leur rôle reste plus ou moins secondaire. La corrélation entre surface et prix total est positive mais modérée, et les analyses montrent que les petites surfaces et les logements avec peu de pièces affichent généralement des prix au m² plus élevés. Toutefois, ces caractéristiques n’expliquent pas à elles seules les variations observées.

Dans l’ensemble, les résultats soulignent que la localisation constitue le facteur le plus structurant du marché immobilier. Celui-ci apparaît fortement segmenté, avec la coexistence de marchés urbains très attractifs et coûteux et de marchés plus accessibles, caractérisés par des prix plus homogènes.

V) Modèles de prédiction

L'objectif de cette partie est d'ajouter un module de prédiction permettant d'estimer le prix d'un appartement à partir de ses caractéristiques. La modélisation repose exclusivement ici sur les annonces d'appartements comme pour notre analyse. Les variables explicatives retenues sont la surface, le nombre de pièces et les indicateurs de localisation. Les variables non numériques mais textuelles ou catégorielles ont été exclus pour éviter de trop complexifier le modèle. Les variables présentant des ordres de grandeurs différentes ont toutes été standardisées pour faciliter l'analyse.

1er Modèle : La Régression Linéaire

Dans un premier temps on utilise le modèle de régression linéaire comme suggéré dans la consigne c'est un modèle assez simple et facilement interprétable. Il va nous permettre d'estimer le prix des appartements à partir de leurs caractéristiques. En premier, le jeu de données est séparé en deux ensemble, un bloc d'entrainement avec 80% des observations, cela va lui servir à apprendre les relations entre les variables explicatives et le prix et un autre bloc avec les 20% restants, qui sont conservées pour tester les performances sur des données qu'il n'a pas vu pendant l'apprentissage. Ensuite on standardise les variables puis le modèle prédit les prix des appartements du jeu de test. Les résultats obtenus sont évalués avec le coefficient de détermination (R^2) qui mesure la part de la variance du prix expliquée par le modèle et du RMSE qui permet d'apprécier l'erreur moyenne de la prédiction en euros. Ainsi on obtient R^2 : 0.228 et RMSE : 168344 €, ce qui signifie qu'environ 22,8% de la variabilité des prix des appartements est expliquée par les variables du modèle. Ce résultat montre certes que ces caractéristiques jouent un rôle dans la formation des prix mais elles ne suffisent pas à expliquer l'ensemble des différences observées entre les biens. Le RMSE élevé s'explique par la dispersion importante des prix et par le nombre limité de variables utilisées.

2ème Modèle : Le Random Forest

Dans un second temps, un modèle de type Random Forest a été testé afin de compléter l'analyse et d'évaluer si une approche plus flexible pouvait améliorer la prédiction des prix. Contrairement à la régression linéaire, ce modèle ne suppose pas une relation strictement linéaire entre les caractéristiques des appartements et leur prix. Il repose sur la combinaison de plusieurs arbres de décision, ce qui lui permet de mieux prendre en compte des relations plus complexes. Le modèle a été entraîné sur le même ensemble d'apprentissage que le modèle de référence, puis utilisé pour prédire les prix des appartements du jeu de test. Les performances obtenues, évaluées à l'aide du R^2 et du RMSE, montrent une amélioration par rapport à la régression linéaire, avec un R^2 de 0,40 et une erreur moyenne de prédiction d'environ 148 000 €. Ces résultats suggèrent que la prise en compte de relations non linéaires permet de mieux représenter la formation des prix, même si les performances restent limitées par le nombre restreint de variables disponibles.

Tableau récapitulatif

Le tableau ci-dessus compare les performances des deux modèles de prédiction testés sur le jeu de données. La régression linéaire, utilisée comme modèle de référence, permet d'expliquer une partie du prix des appartements, mais présente des performances limitées, avec un R^2 de 0,23 et une erreur

moyenne de prédiction relativement élevée. Le modèle Random Forest affiche de meilleures performances, avec un R^2 de 0,40 et un RMSE plus faible, indiquant une amélioration de la qualité des prédictions. Cette différence suggère que les relations entre les caractéristiques des appartements et leur prix ne sont pas strictement linéaires et que l'utilisation d'un modèle plus flexible permet de mieux comprendre la complexité du marché immobilier.

	Modèle	R^2 (test)	RMSE (€)
0	Régression linéaire	0.228	168344
1	Random Forest	0.400	148355

VI) Discussion et Limites

1) Fiabilité des données et biais potentiels

L'ensemble de nos résultats reposent sur des données issues d'un scraping effectué sur un seul site d'annonces immobilières. Le problème est que les données ne viennent que d'un seul site EtreProprio, par conséquent toutes les annonces immobilières existant sur le marché ne sont pas prises en compte. Il est donc possible que les biens sur EtreProprio ne représentent pas bien l'ensemble du marché immobiliers pour les villes concernées, donc cela confirme un biais de représentativité. De plus, la structure même des pages web peut varier d'une annonce à l'autre, ce qui expose l'extraction à des erreurs ou à des informations manquantes, même si ce n'était pas forcément notre cas ici, toutefois, la suppression des annonces régulière fausses aussi les résultats puisque ces biens ne sont plus en vente sur le marché par exemple. Malgré le nettoyage effectué, il peut rester certaines incohérences comme parfois par des valeurs extrêmes (biens très chers ou rares) des surfaces inhabituellement élevées pour des appartements ou des prix au mètre carré très chers ou peu chers, ce qui est fortement susceptible d'altérer l'interprétation des tendances.

Un autre biais tient à la catégorisation du type de bien. En effet, le problème peut venir aussi de la manière de classer les biens (Appartements, Maisons etc). Le filtre peut être imprécis et ne pas inclure certains appartements ou plutôt inclure des biens qui ne sont pas vraiment des appartements, des bien hybrides mal classés. Par ailleurs, le calcul du prix au mètre carré repose sur deux variables susceptibles d'être erronées ou incomplètes, on sait que le calcul du prix au mètre carré repose sur le prix total du bien et la surface du bien. Si par exemple une de ces deux variables est mal renseignée ou fausse alors que le prix au mètre carré sera faux à la fin. La géolocalisation automatisée par Nominatim est aussi une source d'incertitude, puisque certaines villes ne sont pas reconnues ou se voient attribuer des coordonnées imprécises, et aussi nous n'avons tout simplement pas les adresses précises (rues, avenues etc).

Enfin, il semble important de reconnaitre que les variables utilisées dans les annonces (prix, surface, nombre de pièces) ne sont pas suffisantes pour rendre compte de la complexité du marché immobilier. Il existe de nombreux facteurs déterminants du prix qui sont absents dans nos datasets. Notamment la localisation précise (l'environnement, est-il : urbain, à la campagne en centre-ville etc), la qualité des biens (état général du logement, neuf ou ancien, la date de construction, l'état aussi du bâtiment, est-ce que l'appartement est équipé ou non meublés, les potentiels travaux récents, le montant des charges etc). L'absence de ces informations crée un biais de variables omises. Et explique donc pourquoi la relation entre les variables clés de notre dataset reste biaisée. Ainsi, le pouvoir de notre analyse actuelle reste limité et les conclusions tirées ne reflètent qu'une partie de la réalité du marché

2) Améliorations possibles

Afin de rendre l'étude plus fiable et solide nous pouvons envisager plusieurs améliorations possibles. Dans un premier temps améliorer la représentativité à plus grande échelle, donc utiliser plusieurs sites immobiliers, pour varier les données et augmenter notre dataset. Cela permettrait d'avoir un échantillon plus grand et représentatif de l'ensemble du marché ce qui réduit le biais lié

à l'utilisation d'un seul site. Par ailleurs nous pouvons toujours améliorer notre technique et stratégie de scraping, avec un code plus performant pour garantir une meilleure récupération des données par exemple, pour éviter les erreurs ou par exemple tout reprendre en cas d'erreur, rajouter des validations etc. En effet, l'intégration de sources de données multiples permettrait de renforcer la représentativité du dataset, et des procédures de nettoyage plus avancées contribueraient à réduire l'influence des valeurs anormales. Par ailleurs, l'ajout d'informations qualitatives supplémentaires comme la localisation plus précise, l'état du logement ou les équipements disponibles etc offrirait une compréhension plus précise des mécanismes de valorisation des biens et du prix plus globalement.

Pour améliorer la fiabilité de l'étude, il faudrait améliorer aussi le nettoyage. Par exemple mettre en places des méthodes avancées pour mieux identifier et supprimer les valeurs extrêmes (prix très bas, très haut, de même pour les surfaces). Cela rendrait les indicateurs statistiques (moyennes médiane etc) beaucoup plus fiables.

VII) Conclusion

Ce projet a permis de mener une analyse statistique approfondie du marché immobilier à partir de données collectées avec le web scraping. L'ensemble du processus allant de l'extraction des annonces, du nettoyage, de la structuration des données, puis de l'analyse descriptive et visuelle avec les graphiques a mis en lumière la forte hétérogénéité des prix immobiliers en France. Les résultats montrent que la localisation reste le déterminant principal du niveau de prix, avant la surface ou le nombre de pièces par exemple. De plus, certains marchés restent très tendus, comme Paris, Annecy ou Aix-en-Provence, ils exercent malgré eux un effet de distorsion significatif sur les indicateurs globaux. Les visualisations réalisées, notamment les boxplots, l'histogramme et le pairplot, ont permis de mieux comprendre la dispersion des valeurs et les relations entre variables, qui confirme la complexité du marché immobilier local.

Toutefois, ce projet a également mis en évidence les limites inhérentes à l'utilisation de données issues du web scraping à cause du manque d'exhaustivité et hétérogénéité des annonces, mais aussi les erreurs potentielles, ou encore l'absence de variables déterminantes pour l'analyse d'un bien immobilier. Malgré ces contraintes, cette analyse montre une première interprétation du marché étudié.