

Méthodologie

Modèle hédonique de prix des maisons (Québec)

Realtor.ca + DuProprio → nettoyage → ingénierie de variables (règles + LLM + vision) →
régression hédonique log-linéaire à effets fixes

Données extraites le 3 juillet 2026

Résumé

Ce document décrit la méthodologie complète de construction d’un modèle hédonique de prix pour les maisons unifamiliales et maisons en rangée du Québec. Le pipeline combine deux jeux de données d’annonces immobilières, un nettoyage des valeurs aberrantes fondé sur des règles, une ingénierie de variables à quatre origines (règles déterministes, extraction de texte par grand modèle de langage, analyse d’images par modèle de vision, effets fixes géographiques) et une estimation par moindres carrés ordinaires du logarithme du prix avec erreurs-types robustes à l’hétéroscédasticité.

Table des matières

1	Origine des données	1
2	Construction du dataset « maisons »	2
3	Nettoyage des valeurs aberrantes	2
4	Construction des variables	2
4.1	Variable cible	2
4.2	Variables structurelles (règles déterministes)	2
4.3	Aménités (règles par mots-clés bilingues)	3
4.4	Détail des pièces (Realtor)	3
4.5	Signaux de mise en marché	3
4.6	Variables géographiques (effets fixes)	3
4.7	Imputation	3
5	Mécanismes d’IA pour l’ingénierie de variables	3
5.1	Extraction de specs depuis la description — GPT-4o-mini	3
5.2	Attributs visuels de façade — Claude Haiku 4.5 (vision)	4
5.3	Rôle central du JSON structuré strict	4
5.4	Intégration au dataset	4
6	Modèle économétrique	4
6.1	Spécification — hédonique log-linéaire à effets fixes	4
6.2	Double spécification (couverture vs richesse)	5
6.3	Estimation et inférence	5
6.4	Interprétation	5
7	Statistiques et diagnostics	5

8 Limites et biais	5
9 Fichiers produits	6

1 Origine des données

Source	Fichier	Nb	Nature
Realtor.ca	dataset_realtor-....json (161 Mo)	27 302	Annonces MLS via crawler Apify
DuProprio	dataset_duproprio-....json (87 Mo)	7 693	Annonces de vente sans in- termédiaire (Apify)

- **Collecte** : scrapers Apify, extraction le **2026-07-03**. Chaque enregistrement est un objet JSON structuré.
- **Champs communs** : `source`, `url`, `listingId`, `title`, `price/priceNumber`, `currency`, `address{...}`, `property{...}`, `features[]`, `description`, `images[]`, `scrapedAt`.
- **Spécificités** :
 - Realtor : `buildingType`, `storeys`, `yearBuilt`, `parkingSpaces`, `rooms[]` (dimensions par pièce); pas de coordonnées GPS; images sans légende utile.
 - DuProprio : `type`, `levels`, `coordinates` (7 691 présentes); images **avec légendes** (*Kitchen*, etc.); pas de `rooms`.
- **Instantané transversal** (une seule date) : pas de dimension temporelle, donc pas d'effet fixe temporel requis.

Identifiant unique : `uid = source + ":" + listingId` (évite toute collision entre les deux espaces d'identifiants).

2 Construction du dataset « maisons »

1. **Fusion** des deux tableaux JSON (schéma harmonisé).
 2. **Définition de « maison »** (unifamiliales + maisons en rangée) :
 - Realtor : `buildingType` $\in$ {House, Row/Townhouse};
 - DuProprio : `type` $\in$ {Home, Bungalow, Semi Detached, Intergenerational, Country Home, Cottage, Villa, Townhouse};
 - *Exclus* : condos, plex (duplex→quintuplex), terrains/lots, fermes, commercial, maisons mobiles.
 3. **Filtre prix présent** : `priceNumber` non nul et > 0 .
- **19 273 maisons avec prix** (15 470 Realtor + 3 803 DuProprio).

3 Nettoyage des valeurs aberrantes

Les prix extraits contiennent des erreurs d'analyse syntaxique (minimum observé 899 \$, maximum 368 G\$). Règles appliquées :

Variable	Règle	Justification
Prix	garder 50 k\$ – 15 M\$	plage plausible au QC ; retire loyers mal analysés et concaténations
Surface habitable	300 – 20 000 ft ² (sinon → manquant)	valeurs physiquement possibles
Terrain	200 – 5 000 000 ft ² (sinon → manquant)	idem
Chambres / sdb	0–15 / 0–12 (sinon → manquant)	plafonds réalistes
Prix/pied ²	garder P0,5 – P99,5 (≈ 73 –1276 \$/ft ²)	retire les erreurs résiduelles de prix ou surface

→ **19 115 maisons** conservées (158 retirées).

Principe. On *borne* plutôt que de supprimer quand une seule variable est aberrante : la variable est mise à *manquant* puis imputée, pour ne pas perdre l'observation entière.

4 Construction des variables

4.1 Variable cible

$$\ln_price = \ln(priceNumber)$$

La transformation logarithmique rend la distribution symétrique, linéarise les effets multiplicatifs et donne des coefficients interprétables en pourcentage d'impact sur le prix.

4.2 Variables structurelles (règles déterministes)

- **Surface & terrain** : analyse d'unités hétérogènes, tout converti en **pieds carrés** : "1535 sqft", "1,333.2 ft²", "11 561.52 ft²" (séparateurs de milliers variés), "961.6 m²" ($\times 10,7639$).
- **Reconstruction de surface via les pièces (Realtor)** : dimensions "23 ft ,4 in x 13 ft ,8 in" → pieds décimaux → aire, sommées sur toutes les pièces. Calibration sur le recouvrement (ratio médian surface réelle / somme des pièces $\approx 0,98$) pour combler `squareFeet` manquant.
- **Autres** : `beds`, `baths`, `parking`, `storeys/levels`, `age = 2026 – yearBuilt` (avec `age2` pour la courbure), drapeaux `has_*` marquant les valeurs imputées.

4.3 Aménités (règles par mots-clés bilingues)

17 variables indicatrices détectées dans `features[] + description + title` (FR/EN) : piscine, bord de l'eau, garage, foyer, climatisation/ thermopompe, sous-sol fini, cul-de-sac, golf, ski, proximité parc/école/autoroute/ transport, absence de voisin arrière.

4.4 Détail des pièces (Realtor)

Nombre de pièces, nombre de niveaux, chambre principale, sous-sol, bureau, salle familiale, atelier, cellier à vin, solarium, mezzanine, bibliothèque, buanderie.

4.5 Signaux de mise en marché

Nombre de photos (`n_images`, écrêté à 60), présence de visite virtuelle — indicateurs indirects de gamme et de qualité de l'annonce.

4.6 Variables géographiques (effets fixes)

- **Ville** nettoyée (Realtor : retrait du quartier entre parenthèses ; minuscules ; normalisation). Villes avec ≥ 20 annonces $\rightarrow$ indicatrice propre, les rares $\rightarrow$ `__other__` (≈ 194 villes en effets fixes).
- **Type de propriété** harmonisé, regroupé si < 30 annonces.
- **Source** (Realtor/DuProprio) $\rightarrow$ indicatrice captant les différences systématiques entre plateformes.

4.7 Imputation

Valeurs manquantes après nettoyage : imputées par **médiane par type de propriété** (puis médiane globale), **avec un drapeau `has_*`** pour que le modèle distingue valeur observée et valeur imputée.

5 Mécanismes d'IA pour l'ingénierie de variables

Deux couches d'IA extraient des variables impossibles à obtenir par règles : l'une depuis le texte, l'autre depuis les images.

5.1 Extraction de specs depuis la description — GPT-4o-mini

- **But** : transformer la description libre en 20 attributs structurés (état : à rénover $\rightarrow$ neuf ; clé en main ; cuisine/sdb rénovée ; revêtements de sol ; comptoirs quartz/granit ; chauffage ; vue sur l'eau ; revenus locatifs ; grand terrain ; secteur tranquille ; mots-clés de vente).
- **Mécanisme** : appel *Chat Completions* avec `response_format = json_schema strict` (schéma imposé, énumérations fermées) $\rightarrow$ sortie JSON valide garantie, sans analyse fragile. `temperature = 0` (déterminisme). Instruction système : extracteur factuel, uniquement d'après le texte ; absence $\rightarrow$ `false/inconnu/[]`.
- **Ingénierie** : ~ 600 tokens en entrée / ~ 110 en sortie par annonce ; **parallèle** (24 fils), **résumable** (cache `specs.jsonl`, clé `uid`), **réessais** exponentiels sur 429/5xx.
- **Coût / temps** : $\sim \$3$ pour ~ 19 k annonces.

5.2 Attributs visuels de façade — Claude Haiku 4.5 (vision)

- **But** : lire ce que le texte ne dit pas — état visuel réel, attrait extérieur (*curb appeal*), niveau de rénovation, matériaux, aménagement paysager, cadre (urbain/banlieue/rural/bord d'eau/boisé), indice de luxe, qualité de la photo.
- **Sélection d'image** : une photo/annonce = la façade. Realtor $\rightarrow$ 1^{re} photo ; DuProprio $\rightarrow$ photo dont la légende est l'adresse (pas une pièce), par exclusion de mots-clés de pièces.
- **Mécanisme** : *Messages API*, image passée par **URL** (`source.type = url`, sans téléchargement), `output_config.format = json_schema` (12 attributs, énumérations fermées). Instruction système : évaluateur objectif ; scores fondés sur des critères concrets (surfaces, matériaux, aménagement), pas sur un goût personnel.
- **Ingénierie** : ~ 2470 tokens en entrée / ~ 100 en sortie par image ; **parallèle** (16 fils), **résumable** (cache `facade_specs.jsonl`), **réessais** sur 429/5xx/529.
- **Coût / temps** : $\sim \$57$ pour ~ 19 k images.
- **Choix de Haiku** : modèle de vision le moins cher et rapide, suffisant pour de la classification d'attributs concrets.

5.3 Rôle central du JSON structuré strict

Le schéma imposé (types + enum + champs requis + `additionalProperties:false`) fait valider la sortie **au niveau de l'API** : le modèle renvoie exactement les champs attendus, dans le bon domaine de valeurs. On obtient des variables directement tabulables et reproductibles, sans post-traitement de texte libre.

5.4 Intégration au dataset

Les sorties des modèles (clé `uid`) sont converties en variables numériques : booléens $\rightarrow$ 0/1, état $\rightarrow$ score ordinal 0–5, revêtements $\rightarrow$ indicatrices (bois franc, quartz/granit), vue $\rightarrow$ indicatrice eau, etc., avec un drapeau `has_gpt` / `is_exterior` marquant la disponibilité.

6 Modèle économétrique

6.1 Spécification — hédonique log-linéaire à effets fixes

$$\ln(P_i) = \alpha + \beta \ln(\text{surface}_i) + \gamma \ln(\text{terrain}_i) + \sum_k \delta_k X_{ki} + \sum_c \lambda_c \mathbb{I}[\text{ville}_i = c] + \varepsilon_i \quad (1)$$

- **Log-log** sur surface et terrain $\rightarrow$ coefficients = **élasticités** (% de prix par % de surface).
- **Log-niveau** sur le reste $\rightarrow$ un coefficient b vaut $\approx (e^b - 1) \times 100$ % d'impact sur le prix.
- **Effets fixes ville** (λ_c) : absorbent le différentiel de localisation (premier déterminant du prix) ; chaque coefficient = prime/décote de la ville vs la référence.
- **Effets fixes** type de propriété et source.

6.2 Double spécification (couverture vs richesse)

Realtor ne rapporte `squareFeet` que pour ~ 38 % des annonces, d'où deux modèles et des prédictions combinées :

- **Modèle A (riche)** : inclut $\ln(\text{surface})$, estimé sur les biens à surface *mesurée* (réelle ou reconstruite via les pièces). Meilleur ajustement.
- **Modèle B (large)** : sans surface, estimé sur *toutes* les maisons $\rightarrow$ price celles sans surface (beds/baths/terrain/pièces + specs).
- **Prédiction finale** : A si surface mesurée, sinon B.

6.3 Estimation et inférence

- **Moindres carrés ordinaires** sur `ln_price`.
- **Erreurs-types robustes à l'hétéroscédasticité HC3** (les résidus immobiliers sont hétéroscédastiques ; HC3 est conservateur en échantillon moyen).
- **Correction du biais de rétrotransformation** :

$$\hat{P} = e^{\widehat{\ln P}} \times e^{\hat{\sigma}^2/2}$$

(correction normale de Duan) — sinon $e^{\widehat{\ln P}}$ sous-estime systématiquement le niveau de prix.

6.4 Interprétation

Coefficients rapportés en **% d'impact prix** + significativité (p-values sur SE HC3). Effets fixes ville présentés comme primes de localisation.

7 Statistiques et diagnostics

Indicateur	Signification
R^2 / R^2 ajusté	part de variance de <code>ln_price</code> expliquée
Écart-type des résidus (log)	→ erreur typique en % = $(e^\sigma - 1) \times 100$
Erreur absolue médiane	$ \text{prix réel} / \text{prix prédit} - 1 $ médian
% à ± 15 % / ± 25 %	part des prédictions dans la fourchette
<code>resid_pct</code> par annonce	surprix/sous-prix vs modèle → repère aubaines et sur-évaluations

Progression observée : v2 (structurel) $R^2 \approx 0,70 \rightarrow$ v3 (+ pièces/photos) $R^2 \approx 0,72 \rightarrow$ v4 (+ specs GPT) $\rightarrow$ v5 (+ vision) attendu.

8 Limites et biais

- **Endogénéité / non-causalité** : de belles photos ou des specs riches corrélient avec des courtiers haut de gamme et des biens chers → coefficients *prédictifs*, pas causaux.
- **Biais de sélection** : uniquement des biens *en vente* (prix demandé, pas prix de transaction) ; les biens sans surface/pièces sont non aléatoires (annonces moins complètes).
- **Imputation** : les médianes par type réduisent la variance des variables imputées (atténuation possible des coefficients), d'où les drapeaux `has_*`.
- **Extraction LLM** : erreurs/hallucinations possibles ; `temperature=0` + schéma strict + énumérations fermées les limitent, sans validation humaine systématique. La vision juge sur *une seule* photo.
- **Instantané unique** : pas de dynamique de marché ; primes de localisation figées au 2026-07-03.
- **Couverture géographique** : villes < 20 annonces regroupées en `__other__`.

9 Fichiers produits

Fichier	Contenu
<code>hedonic.py</code> , <code>hedonic_v4.py</code>	pipelines reproductibles (nettoyage + modèles)
<code>extract_specs.py</code> , <code>facade_vision.py</code>	extracteurs LLM (texte / vision), parallèles et résu- mables
<code>specs.jsonl</code> , <code>facade_specs.jsonl</code>	sorties LLM brutes (clé uid)
<code>maisons_hedonic*.csv</code>	dataset final + prix prédit + <code>resid_pct</code>
<code>hedonic*_summary.txt</code>	sorties statistiques complètes