Prétraitement des données

Avant de plonger dans l'analyse et la modélisation de données, une étape cruciale dans tout projet de science des données est le prétraitement des données. Le prétraitement vise à préparer les données brutes de manière à ce qu'elles soient prêtes à être explorées et analysées. Cette phase est essentielle pour garantir la qualité et la fiabilité de nos résultats.

Le prétraitement des données consiste en plusieurs tâches, notamment la gestion des valeurs manquantes, la suppression du bruit et la transformation des données. L'objectif est de créer un ensemble de données nettoyé et structuré, prêt à être utilisé pour construire des modèles prédictifs ou pour effectuer des analyses approfondies.

Dans cette section de notre rapport, nous allons passer en revue les différentes étapes de prétraitement des données que nous avons entreprises pour notre projet. Nous expliquerons les raisons derrière chaque étape, les techniques utilisées et les outils mis en œuvre. De plus, nous détaillerons comment ces actions contribuent à améliorer la qualité de nos données et à rendre notre travail d'analyse plus efficace.

Commençons par visualiser quelques lignes de notre Data Frame :

	Identifiant_de_document	Reference_document	1_Articles_CGI	2_Articles_CGI	3_Articles_CGI	4_Articles_CGI	5_Articles_CGI	No_disposition	Date_m
1883842	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	25/
2050544	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	26/
3256050	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	22/
2779968	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	13/
1957212	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1	12/

Traitement des valeurs manquantes

Nous pouvons déjà remarquer plusieurs valeurs manquantes dans plusieurs de nos variables. Vérifions le pourcentage de valeurs manquantes pour chacune des colonnes de notre data frame.

```
Identifiant_de_document      100.00
Reference_document           100.00
1_Articles_CGI               100.00
2_Articles_CGI               100.00
3_Articles_CGI               100.00
4_Articles_CGI               100.00
5_Articles_CGI               100.00
No_disposition                 0.00
Date_mutation                  0.00
Nature_mutation                0.00
Valeur_fonciere                1.10
No_voie                       39.39
B/T/Q                         95.57
Type_de_voie                  41.25
Code_voie                      0.85
Voie                           0.85
Code_postal                    0.85
Commune                        0.00
Code_departement               0.00
Code_commune                   0.00
Prefixe_de_section            95.32
Section                        0.00
No_plan                        0.00
No_Volume                     99.74
1er_lot                       68.84
Surface_Carrez_du_1er_lot     91.48
2eme_lot                      92.63
Surface_Carrez_du_2eme_lot    97.67
3eme_lot                      98.72
Surface_Carrez_du_3eme_lot    99.76
4eme_lot                      99.57
Surface_Carrez_du_4eme_lot    99.94
5eme_lot                      99.80
Surface_Carrez_du_5eme_lot    99.97
Nombre_de_lots                 0.00
Code_type_local               43.26
Type_local                    43.26
Identifiant_local            100.00
Surface_reelle_bati           43.33
Nombre_pieces_principales     43.33
Nature_culture                31.60
Nature_culture_speciale       95.62
Surface_terrain               31.60
dtype: float64
```

Certaines variables (Identifiant_de_document, Reference_document et les Articles_CGI) ont 100% de valeurs manquantes. Pour le moment, nous choisissons de garder les variables suivantes :

'No_disposition', 'Date_mutation', 'Nature_mutation', 'Valeur_fonciere', 'No_voie', 'Type_de_voie', 'Code_voie', 'Voie', 'Code_postal','Commune', 'Code_departement', 'Code_commune', 'Section', 'No_plan', 'Nombre_de_lots', 'Code_type_local', 'Surface_reelle_bati','Nombre_pieces_principales', 'Nature_culture', 'Surface_terrain'

Taille de notre jeu de données après suppression des variables avec trop de valeurs manquantes :
(15125102, 20)
On enlève également les ventes avec des valeurs manquantes dans la variable Valeur foncière car c'est la variable que nous voulons prédire.

Taille de notre jeu de données après suppression des valeurs manquantes de Valeur_fonciere :
(14959015, 20)
Nous supprimons également les lignes de notre data frame avec des valeurs manquantes dans le Code Postal

Taille de notre jeu de données après suppression des valeurs manquantes de Code_postal :
(14830324, 20)
Pour faire une classification de le type de local nous devons également supprimer les valeurs manquantes de cette variables.

Taille de notre jeu de données après suppression des valeurs manquantes de Code_type_local :
(8518444, 20)

Traitement des natures de mutations

Regardons maintenant la variable "Nature_mutation" qui décrit la nature des ventes que nous étudions. Voici les différentes modalités de cette variables :

Modalités de la variable Nature_mutation :
['Vente' 'Vente terrain à bâtir' "Vente en l'état futur d'achèvement"
 'Adjudication' 'Echange' 'Expropriation']

Nombre de vente pour chaque nature de mutation :
Nature_mutation
Vente 8091604
Vente en l'état futur d'achèvement 358116
Adjudication 34312
Echange 29163
Vente terrain à bâtir 4673
Expropriation 576
Name: count, dtype: int64
Nous décidons de garder seulement les ventes pour nettoyer notre data frame.

Taille de notre jeu de données en gardant seulement les ventes :
(8091604, 19)

Duplications des ventes

Après visualisation de nos données on peut se rendre compte que certaines ventes sont dupliquées et représentent plusieurs lignes. Nous avons donc décidé de créer une variable "Adresse" avec "No_voie", "Type_de_voie", "Code_voie", "Voie", "Code_commune" et "Code_departement". Cette variable est regroupée avec la date de la vente et nous supprimons les lignes avec la même date et même adressee.

Taille de notre jeu de données en supprimant les duplications de vente :
(4708687, 13)

Nombre de vente par type local
Code_type_local
1 2134129
2 1263375
3 1047694
4 263489
Name: count, dtype: int64

Les outliers

Nous nous occupons maintenant des outliers. Les outliers peuvent fausser les analyses statistiques, affecter la performance des modèles prédictifs et altérer la visualisation des données. En les identifiant et les gérant correctement, on améliore la qualité des données, et on réduit les risques d'erreurs.

Nous décidons donc de suprimmer les ventes avec une valeurs foncière inférieure ou supérieure aux valeurs limites que nous allons définir.

Nombre de ventes avant suppression des outliers : 4708687

Borne Inférieure
44000.0

Borne Supérieure
554000.0

Nombre de ventes après suppression des outliers : 4012271

Traitement de la date

Nous voulons créer les variables "day", "month" et "year" à partir de la variable "Date_mutation".

Nouvelle structure des variables de notre data frame :
No_disposition int64
Valeur_fonciere float64
Code_postal int64
Commune object
Section object
No_plan int64
Nombre_de_lots int64
Code_type_local int64
Surface_reelle_bati float64
Nombre_pieces_principales float64
Nature_culture object
Surface_terrain float64
day object
month object
year object
dtype: object

	code_commune_INSEE	nom_commune_postal	code_postal	libelle_acheminement	ligne_5	latitude	longitude	code_commune	article	nom_commune	nor
0	1001	L ABERGEMENT CLEMENCIAT	1400	L ABERGEMENT CLEMENCIAT	NaN	46.15	4.93	1.00	L'	Abergement-Clémenciat	L'A
1	1002	L ABERGEMENT DE VAREY	1640	L ABERGEMENT DE VAREY	NaN	46.01	5.43	2.00	L'	Abergement-de-Varey	I
2	1004	AMBERIEU EN BUGEY	1500	AMBERIEU EN BUGEY	NaN	45.96	5.37	4.00	NaN	Ambérieu-en-Bugey	
3	1005	AMBERIEUX EN DOMBES	1330	AMBERIEUX EN DOMBES	NaN	46.00	4.91	5.00	NaN	Ambérieux-en-Dombes	
4	1006	AMBLEON	1300	AMBLEON	NaN	45.75	5.59	6.00	NaN	Ambléon	

233979 individus avec un nom de commune pas trouvé, on les enlève du dataset
nombre final d'individus : 3778292

	No_disposition	Valeur_fonciere	Code_postal	Commune	Section	No_plan	Nombre_de_lots	Code_type_local	Surface_reelle_bati	Nombre_pieces_princip
0	1	45000.00	97400	SAINT DENIS	AH	140	1	2	20.00	
1	1	100000.00	83400	HYERES	CN	259	2	2	52.00	
2	1	102000.00	59310	ORCHIES	D	1099	0	1	54.00	
3	1	297000.00	94300	VINCENNES	C	16	2	2	46.00	
4	1	74000.00	44000	NANTES	EY	28	2	2	36.00	

	No_disposition	Valeur_fonciere	Code_postal	Commune	Section	No_plan	Nombre_de_lots	Code_type_local	Surface_reelle_bati	Nombre_pieces_princip
0	1	45000.00	97400	SAINT DENIS	AH	140	1	2	20.00	
1	1	100000.00	83400	HYERES	CN	259	2	2	52.00	
2	1	102000.00	59310	ORCHIES	D	1099	0	1	54.00	
3	1	297000.00	94300	VINCENNES	C	16	2	2	46.00	
4	1	74000.00	44000	NANTES	EY	28	2	2	36.00	

Ajout d'Open Data

Nous pensons que le niveau de vie d'une commune peut être un facteur important pour prédire la valeur foncière des biens immobiliers. C'est pourquoi nous décidons d'ajouter cette information à notre Data Frame.

	code_commune_INSEE	niveau_vie_commune
0	05047	10021.25
1	26142	10215.00
2	11317	10908.50
3	11384	11485.17
4	30153	11680.00
...	...	...
36567	91526	NaN
36568	74203	NaN
36569	78264	NaN
36570	78606	NaN
36571	78439	NaN

[36572 rows x 2 columns]
Pour certaines communes, le niveau de vie n'est pas renseigné. Nous traitons donc ces valeurs manquantes en les remplaçant par la median du niveau de vie.

Aperçu de notre jeu de données final que nous utiliserons pour la prédiction des valeurs foncières

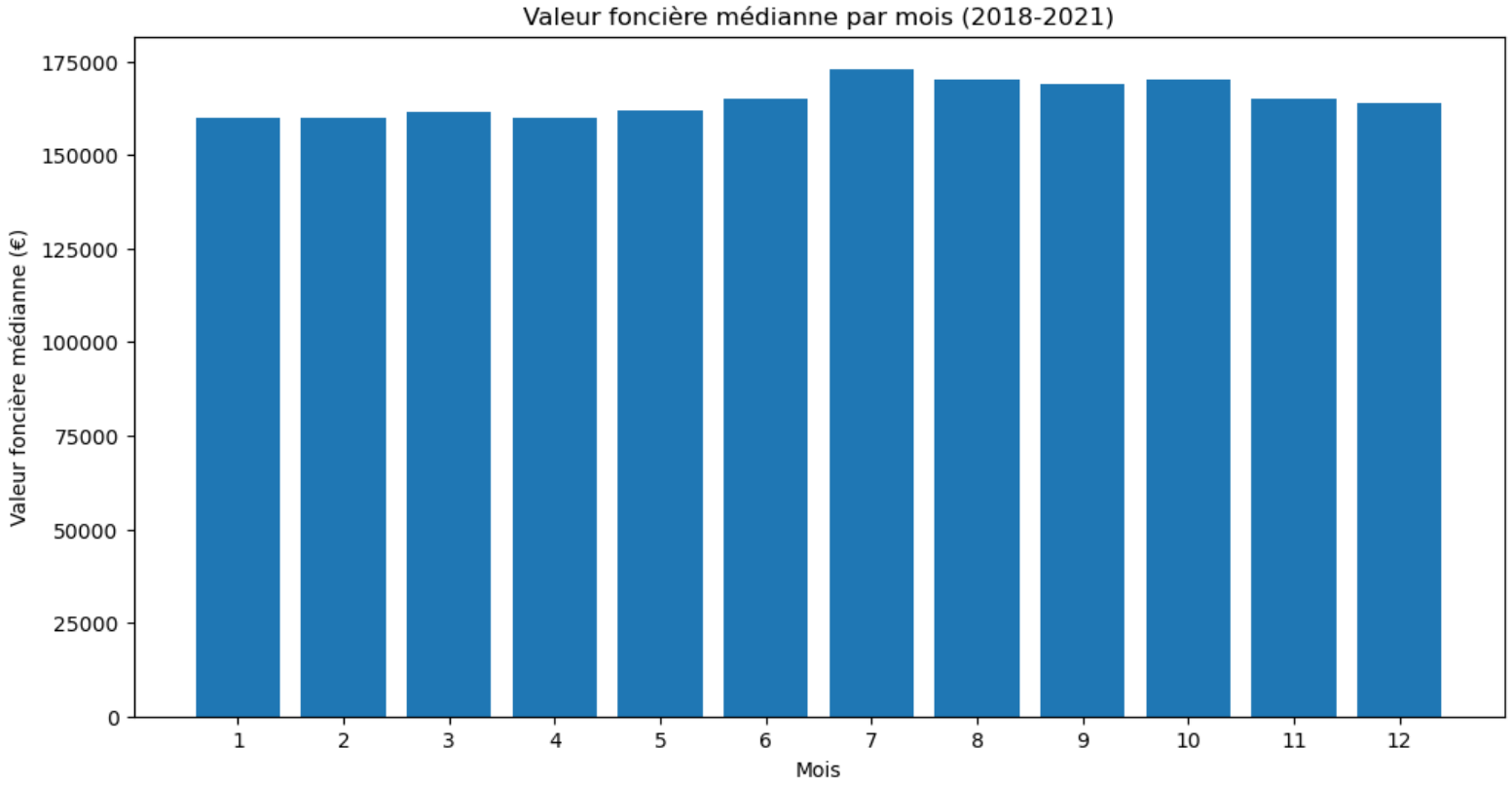
	No_disposition	Valeur_fonciere	Code_postal	Commune	Section	No_plan	Nombre_de_lots	Code_type_local	Surface_reelle_bati	Nombre_pieces_princip
0	1	45000.00	97400	SAINT DENIS	AH	140	1	2	20.00	
1	1	100000.00	83400	HYERES	CN	259	2	2	52.00	
2	1	102000.00	59310	ORCHIES	D	1099	0	1	54.00	
3	1	297000.00	94300	VINCENNES	C	16	2	2	46.00	
4	1	74000.00	44000	NANTES	EY	28	2	2	36.00	

Exploration des données

Avant de procéder à la modélisation de nos variables d'intérêt, il est essentiel d'explorer les données nettoyées afin de rechercher d'éventuelles relations. Ce processus est important car il nous permettra de mieux comprendre les données et, par conséquent, de guider la sélection des variables indépendantes lors de la construction des modèles.

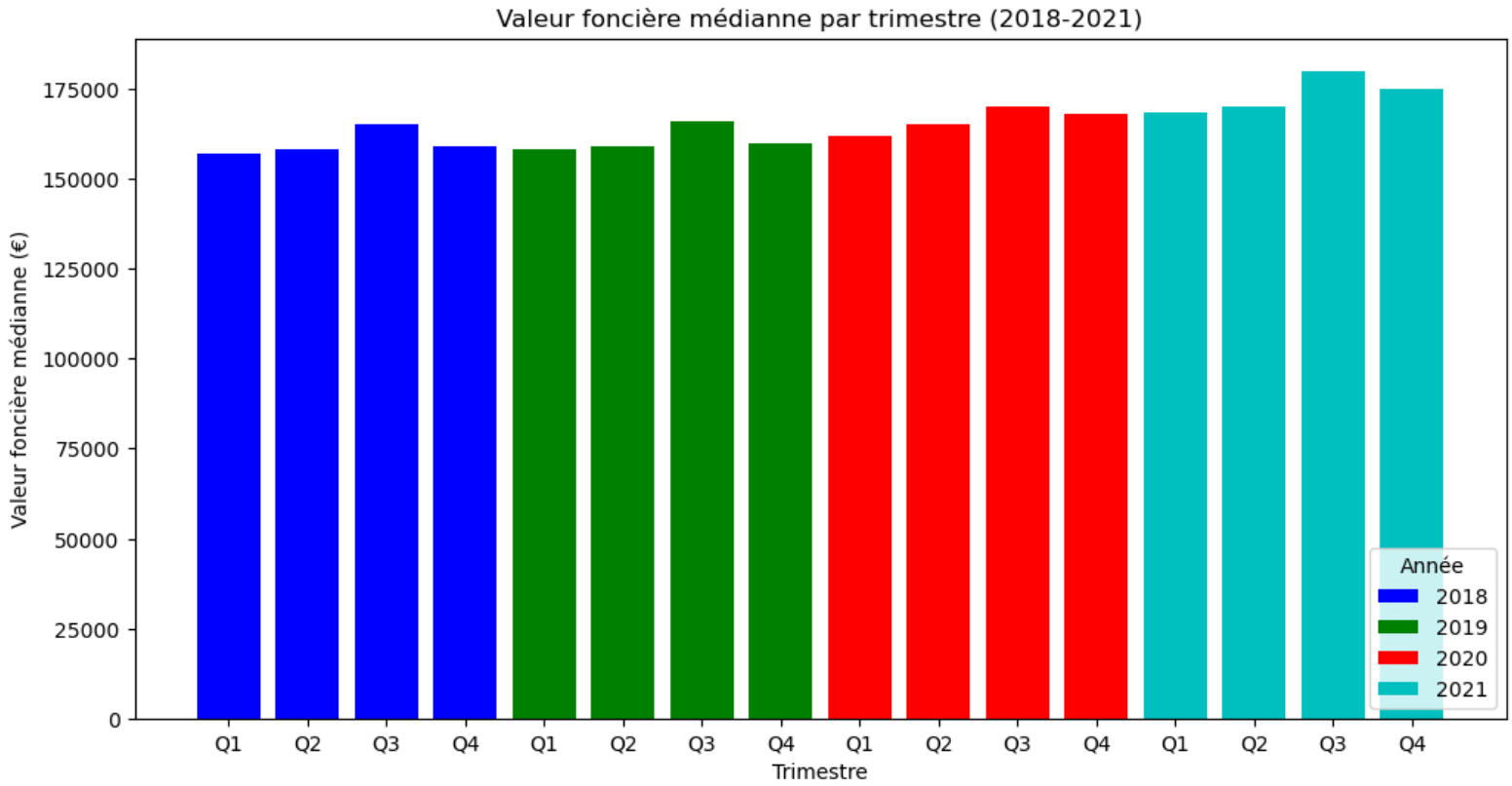
Analyse du temps :

Une variable potentiellement liée à la valeur foncière est le temps. Dans nos données, nous disposons des variables du jour, du mois et de l'année de chaque transaction. On peut donc examiner la relation entre le temps et la valeur foncière de plusieurs manières différentes. Le premier graphique ci-dessous explore la relation entre le « mois » et la valeur foncière médiane correspondante (combinant les données de 2018 à 2021).



Ce graphique démontre qu'il semble y avoir un lien entre le mois de la vente et la valeur foncière, les mois de juillet à octobre semblant avoir une valeur foncière médiane légèrement supérieure aux autres mois.

Il peut également être important d'envisager les changements d'une année à l'autre. Afin de simplifier cela visuellement, nous avons créé une variable « trimestre » pour le graphique suivant, qui montre l'évolution de la valeur foncière médiane d'un trimestre à l'autre entre 2018 et 2021. Ce graphique nous permettra aussi de vérifier si la tendance des mois observée dans la graphique précédante se reproduit pour chaque annéee.



Tout d'abord, ce graphique confirme la tendance du premier graphique. Le troisième trimestre (juillet-septembre) semble avoir une valeur foncière médiane systématiquement plus élevée que les autres trimestres de la même année. De plus, en comparant d'une année à l'autre, il semble y avoir une augmentation graduelle de la valeur financière médiane au fil du temps, avec des valeurs 2021 sensiblement supérieures à celles des années précédentes.

Cette brève analyse démontre l'importance potentielle des variables du temps, tant pour le mois que pour l'année. Ces variables doivent donc être prises en compte lors de la construction des modèles.

Analyse des variables liées au type local

La première partie importante de ce projet consiste à remplacer les données NA pour le Type local avec un modèle de classification. On peut donc explorer la relation entre cette variable à prédire et plusieurs variables indépendantes quantitatives (Le nombre de pièces principales, la surface réelle bati, la surface terrain, le numéro de plan, et la valeur foncière).

Comparaisons des médians

Tout d'abord, le tableau ci-dessous montre les valeurs médianes des cinq variables quantitatives en fonction du type de local (1 - maison, 2 - appartement, 3 - dépendance, 4 - Local industriel, commercial, ou assimilé). Il existe une variation clairement nette entre les valeurs médianes. Il est particulièrement important de noter que la surface réelle du bati (pour les dépendances) et le nombre de pièces principales (pour les dépendances et les locaux industriels/commerciaux) sont égaux à 0. En ce qui concerne les maisons et les appartements, les Les valeurs médianes des variables liées à la taille (nombre de pièces principales, surface réelle bati, et surface terrain) sont plus élevées pour les maisons, ce qui est attendu puisque les maisons sont généralement plus grandes que les appartements. Enfin, la valeur foncière semble également varier selon le type de local, même si cette relation semble moins nette que les autres variables.

Code_type_local	Nombre_pieces_principales	Surface_reelle_bati	Surface_terrain	No_plan	Valeur_fonciere
1	4.0	94.0	500.0	231.0	180000.0
2	2.0	52.0	213.0	171.0	147000.0
3	0.0	0.0	434.0	211.0	160000.0
4	0.0	90.0	570.0	206.0	164000.0

Tests ANOVA

Suite à une analyse préliminaire du tableau des médianes, nous pouvons confirmer ces relations avec un test ANOVA entre le type local et chacune des variables quantitatives. Ce test statistique déterminera s'il existe une différence statistiquement significative entre chacune des modalités de type local (pour chacune des variables indépendantes quantitatives). Les résultats de cet examen sont imprimés ci-dessous, et confirment qu'il existe une variation statistiquement significative entre les modalités de type local pour chaque variable quantitative testée (le nombre de pièces principales, la surface réelle bati, la surface terrain, et la valeur foncière).

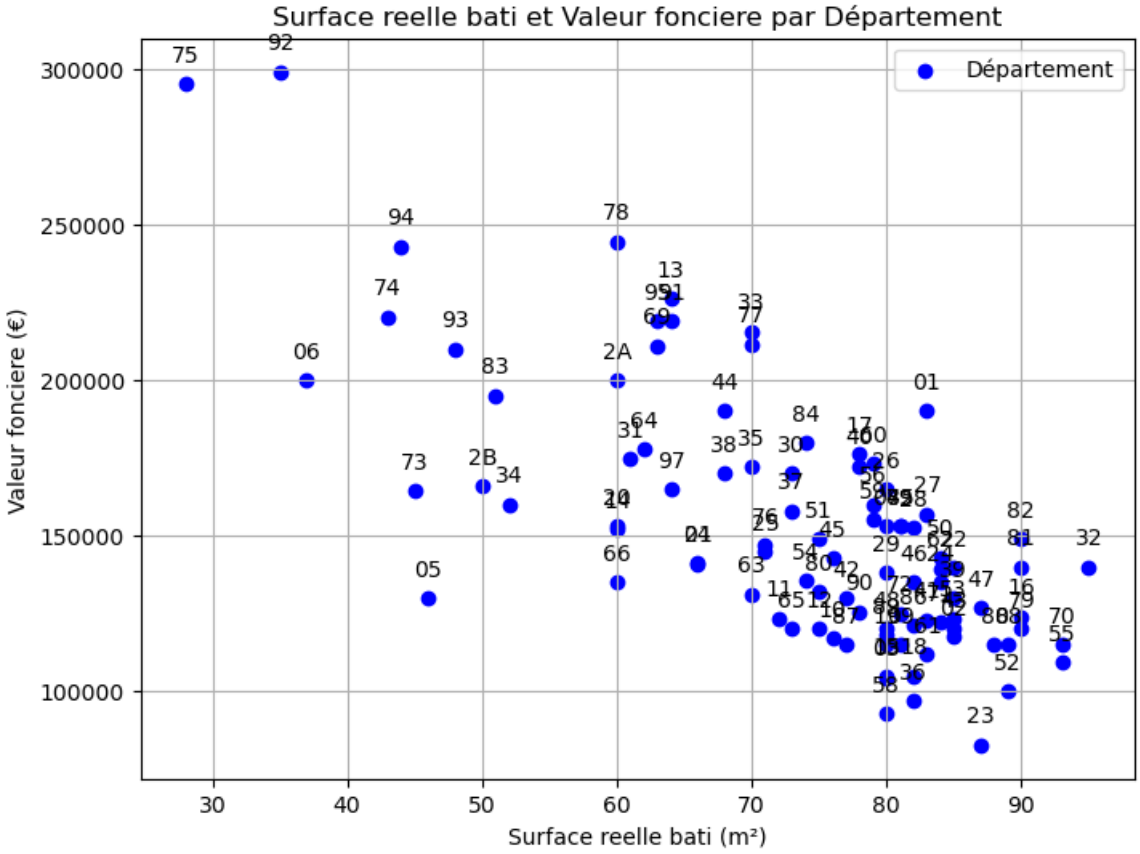
ANOVA pour Nombre_pieces_principales : F-statistic = 832780.4449914864, p-value = 0.0
ANOVA pour Surface_reelle_bati : F-statistic = 30843.71809009009, p-value = 0.0
ANOVA pour Surface_terrain : F-statistic = 4556.348722067259, p-value = 0.0
ANOVA pour Valeur_fonciere : F-statistic = 314.18193151810993, p-value = 5.820534623130433e-204

Analyse des variables liées à la valeur foncière

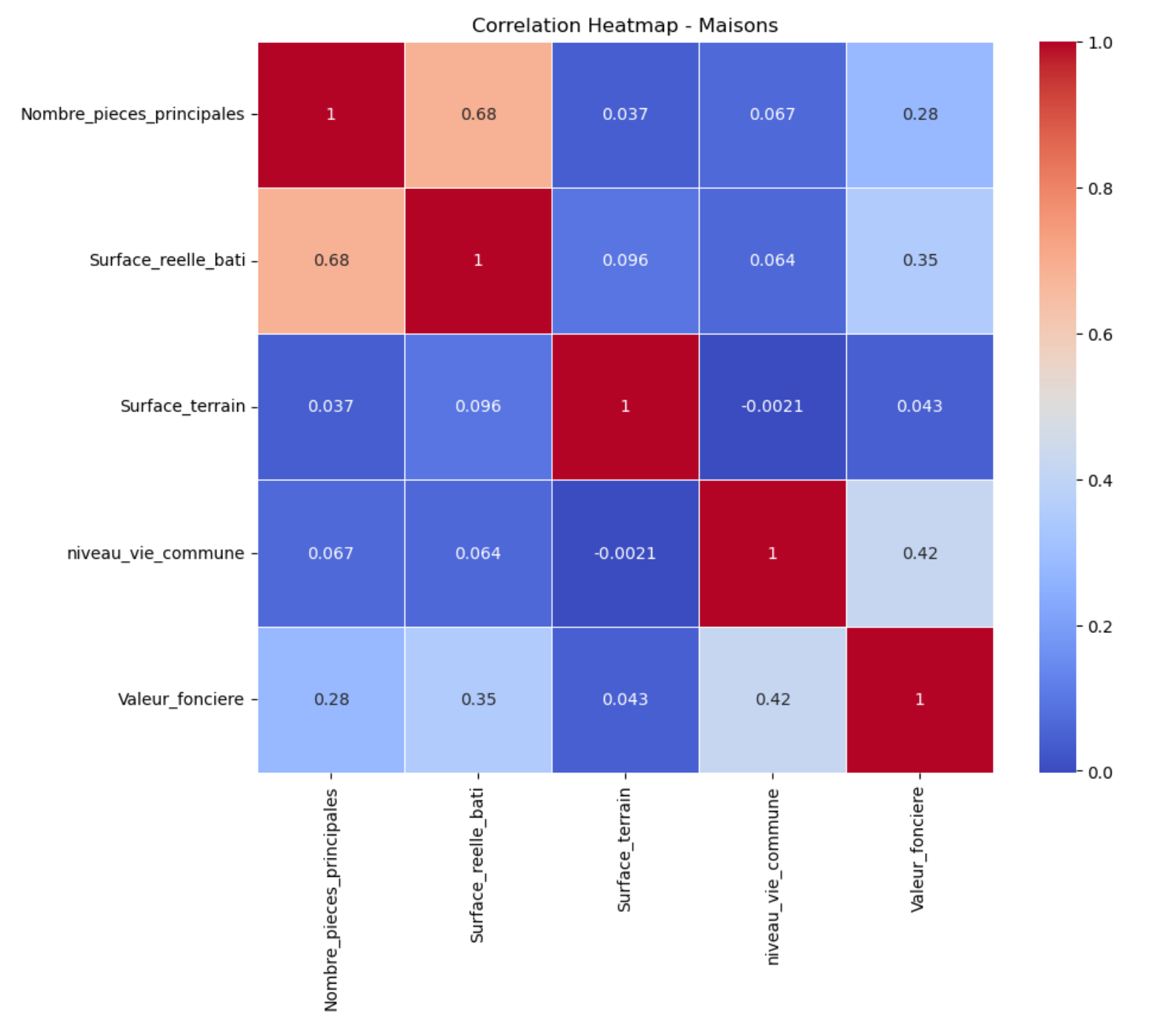
Nous pouvons maintenant nous concentrer sur la variable « Valeur foncière » et sur les variables potentiellement importantes pour le modèle de prédiction des valeurs foncières de 2022. Dans un premier temps, nous examinons les coefficients de corrélation entre la valeur foncière et le nombre de pièces principales, la surface réelle bati, la surface terrain et le niveau de vie de la commune.

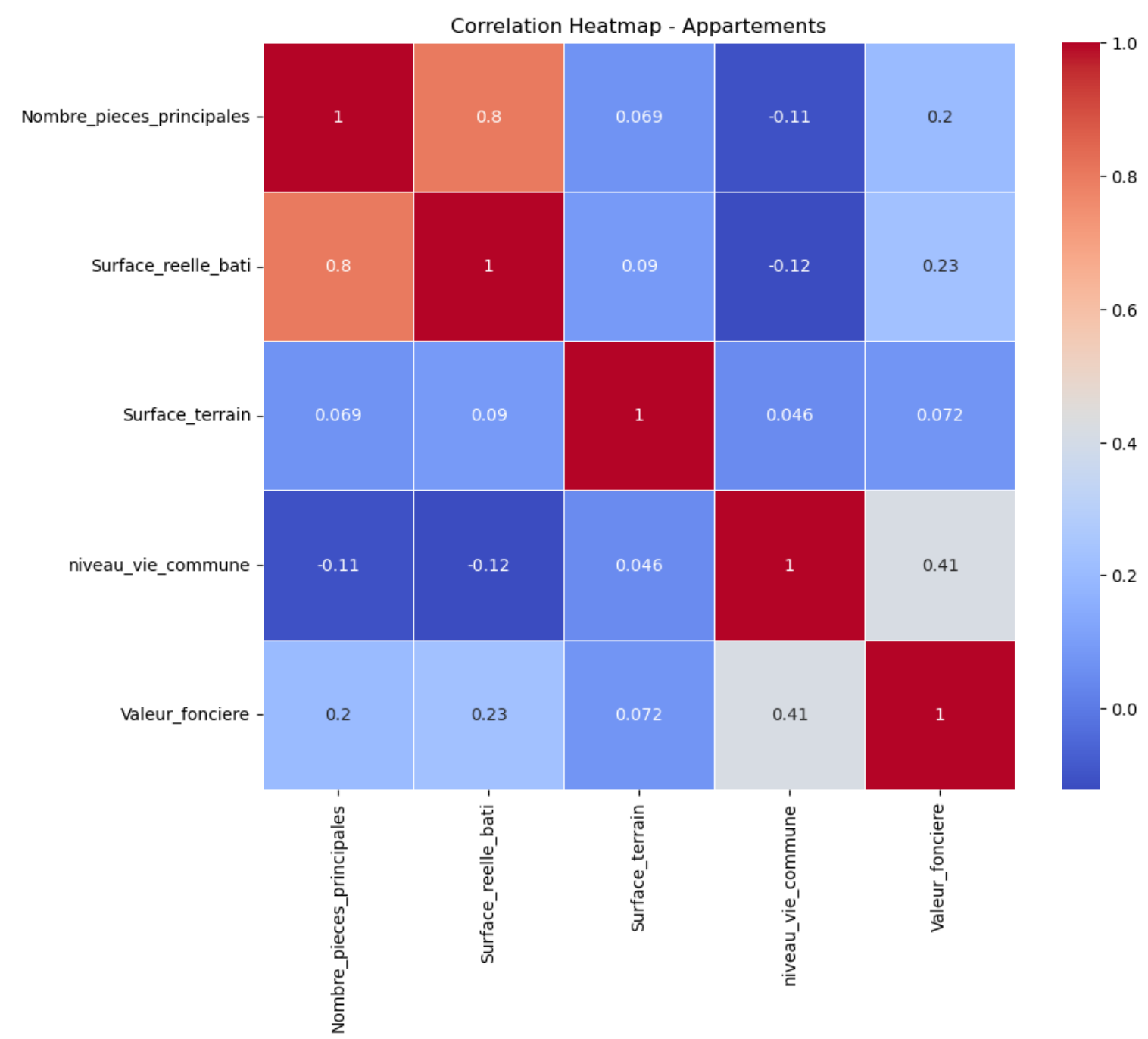
Correlation entre 'Valeur_fonciere' et 'Nombre_pieces_principales': 0.17
Correlation entre 'Valeur_fonciere' et 'Surface_reelle_bati': 0.06
Correlation entre 'Valeur_fonciere' et 'Surface_terrain': 0.05
Correlation entre 'Valeur_fonciere' et 'niveau_vie_commune': 0.36
Les résultats de ces corrélations sont faibles, à l'exception du niveau de vie de la commune. Il pourrait y avoir plusieurs raisons à cela, mais il est probable que la commune/département ainsi que le type de local soient des facteurs contributifs. Le nuage de points ci-dessous montre que certains départements (comme Paris) ont une valeur financière extrêmement élevée, alors qu'une surface réelle moyenne très faible.

En dessous du nuage de points, nous avons calculé le nombre de départements où les maisons ou les appartements sont le type local le plus courant (les deux autres catégories ne sont pas les plus courantes dans aucun département). Ensuite, nous avons imprimé les départements où les appartements étaient plus courants. Beaucoup de ces départements sont des départements urbains, tels que Paris, Lyon ou les banlieues parisiennes. Dans ces départements urbains, le ratio appartements/maisons est beaucoup plus élevé. Il peut donc être intéressant de réexaminer les corrélations en fonction des différentes modalités de type local. Pour tester cette théorie, nous examinerons les cartes thermiques des corrélations pour les maisons et les appartements.



Nombre de départements où 'Maison (1)' ou 'Appartement (2)' est type local la plus fréquent
Code_type_local
1 80
2 15
Name: count, dtype: int64
Les départements où 'Appartement' est le type local le plus fréquent
['05', '06', '20', '2A', '2B', '64', '69', '73', '74', '75', '78', '83', '92', '93', '94']


Correlation Heatmap - Maisons
Nombre_pieces_principales
Surface_reelle_bati
Surface_terrain
niveau_vie_commune
Valeur_fonciere
1
0.68
0.037
0.067
0.28
0.68
1
0.096
0.064
0.35
0.037
0.096
1
-0.0021
0.043
0.067
0.064
-0.0021
1
0.42
0.28
0.35
0.043
0.42
1
1.0
0.8
0.6
0.4
0.2
0.0



Le calcul des corrélations séparées (pour chaque type local) entre les variables indépendantes et la valeur financière donne des résultats plus significatifs pour chacune des variables, sauf pour la surface du terrain. Cela démontre à nouveau l'influence du type local sur la prédiction de la valeur foncière, tout en confirmant que ces variables quantitatives pourraient également être de bons prédicteurs de la valeur foncière.

Classification de la variable "Type local"

Le premier défi de ce projet est de remplacer les valeurs manquantes de la variable « Type local » pour le jeu de données 2022. Pour y parvenir, il est nécessaire de construire et de tester un modèle de classification utilisant les données disponibles de 2018 à 2021, déjà nettoyées.

La partie exploration a révélé que plusieurs variables quantitatives ("Surface_reelle_bati", "Nombre_pieces_principales" et "Surface_terrain") avaient toutes des relations statistiquement significatives avec la variable catégorielle de type local (qui, pour rappel, possède quatre modalités différentes, 1 - Maison, 2 - Appartement, 3 - Dépendance, et 4 - Local industriel, commercial, ou assimilé).

Il est également possible que la commune et d'autres variables géographiques comme la longitude et la latitude puissent jouer un rôle important. Comme nous l'avons vu lors de l'exploration, certains départements (notamment les départements urbains de la métropole parisienne) comptent plus d'appartements que de maisons. Par conséquent, la géographie/localisation peut également jouer un rôle clé dans la prédiction du type local.

Avec ces idées, la première étape consiste à créer un nouveau dataframe qui supprime les colonnes que nous n'utiliserons pas pour la classification, telles que les colonnes liées au temps et d'autres colonnes supplémentaires telles que le code postal. Ce processus nous laisse avec une base de données raffinée qui inclut les variables indépendantes potentielles (ainsi que le type local) pour le modèle de classification. Une fois les variables inutiles pour la classification supprimées, nous pouvons séparer le type local du reste du dataframe.

Imputation des observations manquantes

La dernière étape avant de créer le modèle consiste à s'assurer qu'il n'y a pas de NAN dans les variables indépendantes. Certaines colonnes comportent des observations manquantes. Afin de résoudre ce problème, nous utilisons l'outil SimpleImputer de sklearn, en remplaçant chaque valeur manquante par la médiane de la variable. Une fois l'imputation terminée, nous imprimons le nombre de NAN par colonne pour nous assurer qu'il n'y a plus d'observations manquantes dans les données.

Nombe d'observations NAN
Valeur_fonciere 0
Commune 0
Code_type_local 0
Surface_reelle_bati 0
Nombre_pieces_principales 0
Surface_terrain 0
month 0
latitude 0
longitude 0
niveau_vie_commune 0
dtype: int64

Classification

Modèle 1

Une fois les données séparées et qu'il n'y a plus d'observations manquantes, il est temps de construire le modèle. Pour le premier modèle, nous utiliserons "Surface_reelle_bati" et "Nombre_pieces_principales" comme variables indépendantes. Ce sont les deux variables qui présentaient les relations les plus étroites avec le type local dans les tests ANOVA et constitueront donc un bon point de départ.

Ensuite, nous utilisons la boîte à outils sklearn pour diviser les données en un ensemble d'apprentissage et un ensemble de test (avec 70 % des données dans l'ensemble d'apprentissage). Cela nous permettra de créer un modèle, de l'appliquer pour prédire le type local, et puis de comparer les prédictions avec les valeurs réelles de l'ensemble de test.

Une fois les données divisées, il est temps de créer un modèle de classification à l'aide de l'ensemble d'apprentissage. Nous utilisons un simple classificateur d'arbre de décision pour le premier ensemble de modèles. Une fois que le modèle d'arbre de décision a été créé à l'aide des données d'entraînement, il peut être appliqué à l'ensemble de test pour essayer de prédire le type local de l'ensemble de test. Le score "accuracy" est imprimé, ainsi que l'importance des variables du modèle, le "F1 score" and un tableau croisé entre les observations réelles et les prédictions.

Arbre de décision: 1
['Surface_reelle_bati', 'Nombre_pieces_principales']
Importances : [0.80890466 0.19109534]
Accuracy : 0.8525048968571882
Weighted F1 : 0.8503649835165971

Tableau croisé des obsérvations ('obs') et des valeurs prédites ('pred')

pred	1	2	3	4
obs				
1	0.881712	0.117415	0.000000	0.000873
2	0.320698	0.677869	0.000003	0.001430
3	0.000000	0.000040	0.999960	0.000000
4	0.000000	0.014678	0.035592	0.949730
All	0.511908	0.242899	0.200308	0.044885

Ce premier modèle (avec un score de précision de 85,3%) constitue une première étape très solide. Le modèle est extrêmement efficace pour prédire les dépendances (3) et les Locaux industriels, commerciaux ou assimilés (4). En ce qui concerne les maisons (1) et les appartements (2), cependant, il est un peu difficile de les différencier, avec seulement 2/3 des appartements correctement prédits.

Modèle 2

Par conséquent, nous ajouterons une autre variable qui a également donné des résultats significatifs dans le test ANOVA précédent : "Surface_terrain". Cette variable peut être la clé pour distinguer les maisons des appartements.

Arbre de décision: 2
['Surface_reelle_bati', 'Nombre_pieces_principales', 'Surface_terrain']
Importances : [0.58631766 0.1275066 0.28617574]
Accuracy : 0.9519305087437576
Weighted F1 : 0.9520988705432185

Tableau croisé des obsérvations ('obs') et des valeurs prédites ('pred')

pred	1	2	3	4
obs				
1	0.940160	0.058914	0.000000	0.000926
2	0.060540	0.938024	0.000003	0.001433
3	0.000013	0.000027	0.998979	0.000982
4	0.011731	0.002320	0.031333	0.954616
All	0.469278	0.285474	0.199915	0.045333

L'ajout de la variable "Surface_terrain" a effectivement amélioré la capacité du modèle à distinguer les appartements des maisons. Avec un score de précision amélioré de 95,2 %, le modèle a prédit avec précision 93,8 % des appartements sans perdre la capacité d'attribuer des valeurs correctes aux autres modalités.

Modèle 3

Un modèle avec un score de précision (et un score F1) de 95 % est déjà extrêmement efficace. Cependant, nous n'avons pas encore inclus de variables géographiques. Par conséquent, nous ajouterons simplement la latitude comme autre variable quantitative indépendante. Bien que "Commune" puisse être un prédicteur plus efficace, il serait nécessaire de créer des variables dummy pour chacun des communes (dont il y en a 30 248 dans l'ensemble de données).

Arbre de décision: 3
Variables utilisée dans l'arbre de classification
['Surface_reelle_bati', 'Nombre_pieces_principales', 'Surface_terrain', 'latitude']
Importances : [0.55469589 0.11809309 0.26788001 0.05933101]
Accuracy: 0.9478718523354921
Weighted F1 : 0.9520988705432185

Tableau croisé des obsérvations ('obs') et des valeurs prédites ('pred')

pred	1	2	3	4
obs				
1	0.944319	0.054634	0.000002	0.001046
2	0.080852	0.917983	0.000013	0.001153
3	0.000009	0.000022	0.994377	0.005592
4	0.011408	0.006788	0.019716	0.962088
All	0.476828	0.278130	0.198466	0.046577

L'inclusion de la latitude n'a pas permis d'améliorer le modèle et a en fait eu un impact négatif sur la capacité d'identifier correctement les appartements. Il est donc préférable d'exclure "latitude" comme variable, et de revenir aux trois variables du deuxième modèle (Surface_reelle_bati, Nombre_pieces_principales et Surface_terrain).

Modèle 4

Avec ces variables, nous pouvons tenter d'utiliser une méthode de classification différente pour voir si nous pouvons améliorer légèrement les résultats, même si une précision de 95 % est déjà un indicateur d'un modèle de haute qualité. Les modèles de Random Forrest peuvent améliorer la précision des modèles d'arbres de décision classiques en combinant plusieurs arbres. Ainsi, avec les mêmes variables indépendantes, nous pouvons faire un modèle de Random Forrest (en suivant la même procédure de préparation que pour l'arbre de décision).

Random Forrest: 1
['Surface_reelle_bati', 'Nombre_pieces_principales', 'Surface_terrain']
Accuracy: 0.9528375302193439
Weighted F1 : 0.9520988705432185

Tableau croisé des obsérvations ('obs') et des valeurs prédites ('pred')

pred	1	2	3	4
obs				
1	0.940807	0.058197	0.000000	0.000996
2	0.059094	0.939460	0.000003	0.001443
3	0.000013	0.000027	0.998907	0.001053
4	0.007985	0.001464	0.031276	0.959274
All	0.469019	0.285482	0.199899	0.045600

Le modèle de Random Forrest a donné des résultats pratiquement identiques au meilleur modèle d'arbre de décision. Cependant, la production du modèle a pris près de cinq minutes et n'est donc pas idéal, en particulier pour le plus grand base de données de 2022. Ainsi, le deuxième modèle (l'arbre de décision à trois variables indépendantes : Surface_reelle_bati, Nombre_pieces_principales et Surface_terrain), est le modèle que nous utiliserons pour prédire les observations manquantes de 2022.

Nous exporterons donc ce modèle afin de pouvoir l'appliquer aux données de 2022.

Régression pour la prédiction des valeurs foncières

Après avoir fait notre modèle de classification sur la variable type local, nous passons à la réalisation de note modèle de régression.

Un modèle unique pour tous les types de local

Une de nos première piste de travail est l'utilisation d'ElasticNet. Ce modèle offre plusieurs avantages pour la prédiction. C'est une technique d'apprentissage automatique qui combine les approches de régression Lasso (L1) et Ridge (L2) en une seule méthode.

Nous décidons de créer un premier modèle en passant plusieurs paramètre dans un objet GridSearchCV. Cet objet est créé pour effectuer une recherche sur la grille des meilleurs paramètres pour le modèle Elastic Net. La recherche est effectuée en utilisant une validation croisée à 5 plis et en évaluant la métrique de l'erreur quadratique moyenne négative (neg_mean_squared_error).

Le GridSearch va nous permettre de tester notre modèle avec différentes valeurs pour les paramètres l1_ratio ainsi que alpha.

L'hyperparamètre l1, également appelé l1_ratio, contrôle le mélange relatif de la régression L1 (Lasso) et L2 (Ridge) dans le modèle Elastic Net. Il varie entre 0 et 1.

L'hyperparamètre alpha contrôle le niveau global de régularisation appliqué au modèle Elastic Net. Il varie entre 0 et l'infinie. En augmentant alpha, on augmente la pénalisation des coefficients du modèle, ce qui peut aider à prévenir le surajustement en rendant le modèle plus simple.

Variables utilisées dans notre régression :

```
Index(['Code_type_local', 'Surface_reelle_bati', 'Nombre_pieces_principales',
       'Surface_terrain', 'month', 'latitude', 'longitude',
       'niveau_vie_commune', 'Prix_moyen_m2'],
      dtype='object')
```

Meilleurs paramètres pour Elastic Net: {'alpha': 0.1, 'l1_ratio': 0.9}

Erreur Quadratique Moyenne (MSE) : 95026.04352331463

```
         Valeur_reelle  Valeur_predite
1993658       259000.0   253948.240587
306571        140000.0   148824.378048
2195275        93500.0   116729.460858
450558        177150.0   189512.748303
74297          50000.0   137970.755559
...                ...            ...
2516661        70000.0   166364.489693
1135889        75000.0   158178.860884
2766961       110000.0   190415.531220
1604147       391800.0   245468.764877
1627862        90000.0   177823.684360

[755574 rows x 2 columns]
```

R carré : 0.2578877953297367
Le RMSE ainsi que le R^2 montre que le modèle n'est pas très performant sur nos données.

Un modèle par type de local

Une deuxième piste pour notre modèle est de l'entrainer pour chaque type local afin de gagner en précision. Pour cela, nous voulons tester la régression Ridge, Lasso et ElastiNet en faisant varier leurs paramètres alpha et l1_ratio. Nous utilisons toujours GridSearch pour tester chacun des modèles.

Voici les résultats par modèles et par code type local :

	ElasticNet alpha=10.0 l1_ratio=0.1	ElasticNet alpha=10.0 l1_ratio=0.5	ElasticNet alpha=10.0 l1_ratio=0.9	ElasticNet alpha=0.1 l1_ratio=0.1	ElasticNet alpha=0.1 l1_ratio=0.5	ElasticNet alpha=0.1 l1_ratio=0.9	ElasticNet alpha=1.0 l1_ratio=0.1	ElasticNet alpha=1.0 l1_ratio=0.5	ElasticNet alpha=1.0 l1_ratio=0.9
Code_type_local 1	75036.631536	75019.953988	74983.682996	74966.945473	74966.551444	74966.289065	74981.822778	74974.181542	74967.065549
Code_type_local 2	77657.221582	77646.324176	77632.700600	77649.855086	77655.245544	77662.446005	77632.417481	77633.004169	77648.705015
Code_type_local 3	100118.598356	100107.729550	100097.749655	100096.051314	100095.888945	100095.679459	100097.545383	100096.904581	100096.093128
Code_type_local 4	120638.097557	120631.160335	120602.261031	120492.185710	120471.071321	120447.318030	120598.935504	120577.235056	120496.870802

Le code 1 correspond aux maisons, le 2 aux appartements, le 3 aux dépendances, et le 4 aux locaux. On peut voir que les performances de chaque modèle ne varie pas énormément en fonction des hyperparamètres.

On choisit donc les meilleurs modèles avec leurs hyperparamètres correspondants.

Voici les modèles sélectionnés par code type local ainsi que leurs performances :

```
{1: {'Modèle': Ridge(alpha=0.1), 'RMSE': 74966.24784199636, 'Alpha': 0.1}, 2: {'Modèle': ElasticNet(l1_ratio=0.1), 'RMSE': 77632.41748059462, 'Alpha': 1.0, 'L1_ratio': 0.1}, 3: {'Modèle': Ridge(alpha=0.1), 'RMSE': 100095.61674065409, 'Alpha': 0.1}, 4: {'Modèle': Ridge(alpha=0.1), 'RMSE': 120442.01557765303, 'Alpha': 0.1}}
```

Au final, les variables utilisées dans nos modèles sont :

```
Index(['Surface_reelle_bati', 'Nombre_pieces_principales', 'Surface_terrain',
       'month', 'latitude', 'longitude', 'niveau_vie_commune',
       'Prix_moyen_m2'],
```