Afficher le code

PRÉPARATION DES DONNÉES

Inspection des données

Afficher le code

Avant de pouvoir lancer nos modèles de prédiction de Valeur foncière, nous procédons avant tout à l'inspection des données.

Afficher le code

Afficher la sortie masquée

On remarque qu'il y a beaucoup de valeurs manquantes. Il sera donc nécessaire de nettoyer les données.

Afficher le code

```
Voici les dimensions du dataset complet : (15125102, 43)
```

Nettoyage des données

Après avoir inspecté les données, nous procédons au nettoyage de celles-ci.

Afficher le code

```
Après élimination des duplicats, le nombre d'observations se réduit à : (3738104, 43)
```

Nous inspectons les valeurs manquantes.

Afficher le code

On remarque que beaucoup de variables ont un taux de valeurs manquantes supérieur ou égal à 80%.

Afficher le code

Afficher la sortie masquée

Nous décidons donc d'éliminer les variables pour lesquelles le taux de valeurs manquantes excède ou est égal à 80%.

Afficher le code

Afficher la sortie masquée

Après filtrations des valeurs manquantes, les données sont réduites.

Afficher le code

```
Après filtrage global des valeurs manquantes, il reste 22 variables : (3738104, 22)
Après avoir filtré les valeurs manquantes de Type local, les données sont réduites à : (2342692, 22)
```

Nous reformatons Valeur foncière en variable quantitative en remplaçant les "," par des ".".

Afficher le code

Afficher la sortie masquée

Cela nous permet ensuite d'éliminer les valeurs aberrantes de Valeur foncière (i.e., inférieure à 1 ou NaN).

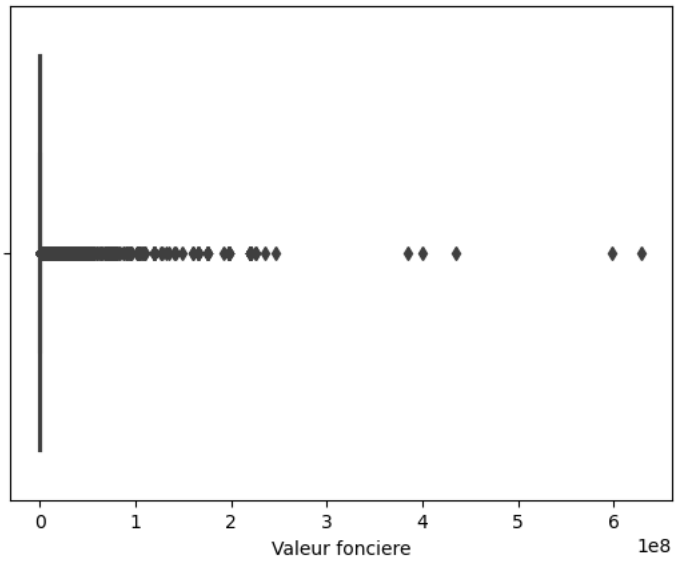
Afficher le code

```
Avant filtrage des valeurs aberrantes de Valeur foncière : (2337078, 22)
Après filtrage des valeurs aberrantes de Valeur foncière : (2337078, 22)
```

Nous éliminons ensuite les outliers restant de Valeur foncière (cf. boxplot ci-dessous).

Afficher le code

```
<Axes: xlabel='Valeur fonciere'>
```



Nous définissions d'abord les premier et troisième quartile. Nous calculons l'écart interquartile afin d'obtenir les bornes inférieure et supérieure d'exclusion des outliers. Et nous les éliminons selons ces critères.

Afficher le code

```
Premier quartile : 85000.0
Troisième quartile : 250000.0
Écart interquartile : 165000.0
Borne inférieure d'exclusion : -162500.0
Borne supérieure d'exclusion : 497500.0
Dataset avant exclusion des outliers : (2337078, 22)
Dataset après exclusion des outliers : (2204117, 22)
```

Nous décidons de nous concentrer essentiellement sur les ventes (cf. variable "Nature mutation"), ce qui réduit encore le nombre de données exploitables.

Afficher le code

```
Dataset après filtrage des ventes : (2166999, 22)
```

La variable "Date mutation" n'étant pas intéressante pour nos modèles de prédiction, nous décidons de l'éliminer.

Afficher le code

```
Nombre de variables après exclusion de Date mutation : (2166999, 21)
```

Nous devons ensuite reformater les variables "Code departement" et "Code voie" en variables catégorielles.

Afficher le code

Afficher la sortie masquée

Nous remplaçons les valeurs manquantes des variables quantitatives par leurs moyennes. Et nous vérifions qu'il ne reste plus de valeurs manquantes dans les variables quantitatives.

Afficher le code

Afficher la sortie masquée

Nous décidons de ne garder que les variables où il n'y a plus de valeurs manquantes.

Afficher le code

```
Sur 21 variables, il n'en reste donc plus que 16 : (2166999, 16)
Voici la liste des variables restantes : Index(['No disposition', 'Nature mutation', 'Valeur fonciere', 'No voie',
       'Code voie', 'Code postal', 'Commune', 'Code departement',
       'Code commune', 'No plan', 'Nombre de lots', 'Code type local',
       'Type local', 'Surface reelle bati', 'Nombre pieces principales',
       'Surface terrain'],
      dtype='object')
```

On affiche une matrice de corrélations des 16 variables restantes.

Afficher le code

Afficher la sortie masquée

Afficher le code

```
Type local                      -0.100590      0.124993  0.102470
Surface reelle bati             -0.026515      0.024093  0.017790
Nombre pieces principales       -0.068642      0.102598  0.057566
Surface terrain                 -0.011193     -0.002091 -0.000391

                        Nombre de lots  Code type local  Type local  \
No disposition               -0.061327        -0.033399    0.063804
Nature mutation                    NaN              NaN         NaN
Valeur fonciere              -0.089796        -0.240355    0.193589
No voie                       0.004763         0.134677   -0.051498
Code voie                     0.153039         0.158021   -0.198994
Code postal                   0.061610         0.037786   -0.059365
Commune                       0.007099         0.003074   -0.009565
Code departement              0.092256         0.069353   -0.100590
Code commune                 -0.100185        -0.091865    0.124993
No plan                      -0.088465        -0.064808    0.102470
Nombre de lots                1.000000         0.498317   -0.723246
Code type local               0.498317         1.000000   -0.616415
Type local                   -0.723246        -0.616415    1.000000
Surface reelle bati          -0.075021         0.016328    0.110146
Nombre pieces principales    -0.361506        -0.763481    0.511065
Surface terrain              -0.000441         0.063970   -0.009419

                        Surface reelle bati  Nombre pieces principales  \
No disposition                     0.027976                   0.031007
Nature mutation                         NaN                        NaN
Valeur fonciere                    0.095723                   0.370190
No voie                            0.012459                  -0.136813
Code voie                         -0.025630                  -0.164934
Code postal                       -0.019099                  -0.033035
Commune                           -0.002503                  -0.008663
Code departement                  -0.026515                  -0.068642
Code commune                       0.024093                   0.102598
No plan                            0.017790                   0.057566
Nombre de lots                    -0.075021                  -0.361506
Code type local                    0.016328                  -0.763481
Type local                         0.110146                   0.511065
Surface reelle bati                1.000000                   0.082990
Nombre pieces principales          0.082990                   1.000000
Surface terrain                    0.230246                  -0.005242

                        Surface terrain
No disposition                 0.012474
Nature mutation                     NaN
Valeur fonciere                0.036524
No voie                        0.056253
Code voie                      0.048241
Code postal                   -0.009735
Commune                        0.003773
Code departement              -0.011193
Code commune                  -0.002091
No plan                       -0.000391
Nombre de lots                -0.000441
Code type local                0.063970
Type local                    -0.009419
Surface reelle bati            0.230246
Nombre pieces principales     -0.005242
Surface terrain                1.000000
```

Sélection des variables intéressantes

Après avoir inspecté et nettoyé les données, nous devons à présent sélectionner les variables que nous utiliserons pour nos modèles de prédiction.

Nous standardisons toutes les variables quantitatives (hormis Valeur foncière et Code commune) afin de les ramener à la même échelle.

Afficher le code

Afficher la sortie masquée

Afficher le code

Afficher la sortie masquée

Nous lançons une régression Lasso afin de déterminer quelles variables pèsent le plus sur Valeur foncière. Ici, nous affichons ces variables lorsque leurs coefficients dépassent un certain seuil.

Afficher le code

Afficher la sortie masquée

Nous affichons ensuite les 10 variables les plus importantes parmi celles sélectionnées précédemment.

Afficher le code

```
Commune
No voie
Code departement
Surface reelle bati
Code voie
Nombre de lots
Code postal
Code type local
Type local
Nombre pieces principales
```

Maintenant que les données ont été inspectées et nettoyées, et que les variables les plus importantes ont été mises en exergue, nous pouvons passer aux modèles de prédiction.

DATAFRAME SANS TRANSFORMATIONS

Nous récupérons un dataset sans transformation des données afin de pouvoir lancer nos modèles de prédiction (cf. le split train-test ne peut être lancé sur des données préalablement transformées).

Afficher le code

Afficher la sortie masquée

Nous pouvons ainsi lancer les modèles de classification afin de prédire Type local, qui nous aideront ensuite à prédire Valeur foncière.

CLASSIFICATION

Échantillons d'apprentissage vs. Échantillons test

Nous définissons nos échantillons d'apprentissage et de test.

Pour cela, nous décidons de ne garder que 4 variables : après des analyses préalables (non visibles ici), nous avons constaté l'enrichissement des modèles prédictifs seulement pour les variables "Surface reelle bati", "Nombre de lots", et "Nombre pieces principales" dans la prédiction de "Type local".

Afficher le code

Nous définissons nos variables explicatives : "Surface reelle bati", "Nombre de lots", "Nombre pieces principales" ; et notre variable cible : "Type local".

Afficher le code

Nous vérifions la distribution des modalités de Type local.

Afficher le code

Afficher la sortie masquée

Nous effectuons notre split échantillons d'apprentissage vs. échantillons test. Nous vérifions que les distributions de Type local sont bien les mêmes entre apprentissage et test.

Afficher le code

```
Distribution de Type local en apprentissage :
Type local
Maison                                  0.575961
Appartement                             0.273039
Dépendance                              0.100647
Local industriel. commercial ou assimilé    0.050352
Name: proportion, dtype: float64
Distribution de Type local en test :
Type local
Maison                                  0.575961
Appartement                             0.273040
Dépendance                              0.100648
Local industriel. commercial ou assimilé    0.050352
Name: proportion, dtype: float64
```

Transformation des données

Pour l'optimisation temporelle des modèles, nous avons fait le choix de remplacer les valeurs manquantes par la moyenne (pour les variables quantitatives explicatives).

Notre premier choix était de remplacer les valeurs manquantes par les kNN. Cependant, cela prenait trop de temps dans le traitement.

Afficher le code

Afficher la sortie masquée

Nous pouvons de nouveau standardiser les variables explicatives en apprentissage et en test.

Afficher le code

Afficher la sortie masquée

Afficher le code

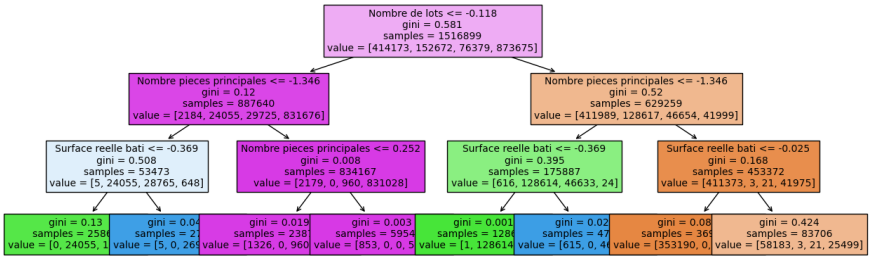
Afficher la sortie masquée

Nous lançons nos modèles prédictifs de Type local.

Arbre de décision

Nous laçons un arbre de décision pour prédire Type local.

Afficher le code



À l'aide de notre modèle, nous prédisons Type local.

Afficher le code

```
Voici la matrice de confusion entre les valeurs de Type local observées vs. p
```

pred obs	Appartement	Local industriel. commercial ou assimilé	Maison	All
Appartement	176273	280	950	177503
Dépendance	0	65430	1	65431
Local industriel. commercial ou assimilé	8	32325	401	32734
Maison	18196	279	355957	374432

Nous évaluons ensuite notre modèle prédictif grâce à différentes métriques.

Afficher le code

```
/Users/annabellenarsama/anaconda3/envs/env/lib/python3.11/site-packages/sklearn/metrics/_classification.py:1344: Undefin
  _warn_prf(average, modifier, msg_start, len(result))
/Users/annabellenarsama/anaconda3/envs/env/lib/python3.11/site-packages/sklearn/metrics/_classification.py:1344: Undefin
  _warn_prf(average, modifier, msg_start, len(result))
/Users/annabellenarsama/anaconda3/envs/env/lib/python3.11/site-packages/sklearn/metrics/_classification.py:1344: Undefin
  _warn_prf(average, modifier, msg_start, len(result))
Rapport complet des métriques :
                                    precision    recall  f1-score   support

                       Appartement       0.91      0.99      0.95    177503
                        Dépendance       0.00      0.00      0.00     65431
Local industriel. commercial ou assimilé       0.33      0.99      0.49     32734
                            Maison       1.00      0.95      0.97    374432

                          accuracy                           0.87    650100
                         macro avg       0.56      0.73      0.60    650100
                      weighted avg       0.84      0.87      0.84    650100

L'accuracy est de 0.86841255191509
/Users/annabellenarsama/anaconda3/envs/env/lib/python3.11/site-packages/sklearn/metrics/_classification.py:1344: Undefin
  _warn_prf(average, modifier, msg_start, len(result))
La precision est de 0.8378184647234126
Le recall est de 0.86841255191509
Le f1_score est de 0.843969556153508
```

L'optimisation des hyperparamètres de l'arbre de décision via un GridSearch avec validation croisée n'a pas de sens. En effet, des transformations de standardisation ont été opérées sur les données avant le split train-test de la cross-validation du GridSearch.

KNeighbors Classifier

Nous laçons ensuite un modèle KNeighbors Classifier pour prédire Type local.

Afficher le code

```
▾        KNeighborsClassifier
KNeighborsClassifier(n_neighbors=82)
```

À l'aide du modèle, nous prédisons Type local.

Afficher le code

```
Voici la matrice de confusion entre les valeurs de Type local observées vs. p
```

pred obs	Appartement	Dépendance	Local industriel. commercial ou assimilé	Maison	All
Appartement	172628	5	295	4575	177503
Dépendance	0	65404	27	0	65431
Local industriel. commercial ou assimilé	8	733	31993	0	32734

Nous évaluons notre modèle à l'aide de différentes métriques.

Afficher le code

```
Rapport complet des métriques :
                                    precision    recall  f1-score   support

                        Appartement       0.93      0.97      0.95    177503
                         Dépendance       0.99      1.00      0.99     65431
Local industriel. commercial ou assimilé       0.98      0.98      0.98     32734
                            Maison       0.99      0.97      0.98    374432

                          accuracy                           0.97    650100
                         macro avg       0.97      0.98      0.98    650100
                      weighted avg       0.97      0.97      0.97    650100

L'accuracy est de 0.9722273496385172
La precision est de 0.9728175278750519
Le recall est de 0.9722273496385172
Le f1_score est de 0.9723520849516789
```

L'optimisation des hyperparamètres de ce modèle est non seulement long à lancer, mais en plus n'a pas de sens (les données ont été prélablement transformées avant le split train-test de la validation croisée du GridSearch).

Conclusion - Classification : Notre meilleur modèle de prédiction est le KNeighbors Classifier. Nous garderons donc les résultats de Type local de ce modèle pour prédire Valeur foncière.

OPEN DATA

Nous avons décidé d'ajouter le niveau de vie (par commune) et le prix moyen au m2 à nos données de base afin de prédire Valeur foncière. En effet, dans nos analyses préalables, elles enrichissent les modèles prédictifs.

Afficher le code

Afficher le code

Afficher le code

RÉGRESSION

On repart d'un dataset sans transformations préalables des données.

Afficher le code

Afficher la sortie masquée

On agrège notre dataset avec les open data.

Afficher le code

Échantillons d'apprentissage vs. test

Nous effectuons notre split train-test avant de lancer notre modèle prédictif. Pour cela, nous définissons d'abord nos variables explicatives (Surface réelle bâti, Nombre de lots, Nombre de pièces principales, Type local, open data) et notre variable cible (Valeur foncière).

Afficher le code

Afficher le code

Afficher la sortie masquée

Afficher le code

Transformation des données

Nous pouvons de nouveau procéder aux transformations des données.

Nous inspectons d'abord les valeurs manquantes.

Afficher le code

```
Surface reelle bati          1415
Nombre de lots                  0
Nombre pieces principales    1415
Type local                      0
dtype: int64
0
```

Nous remplaçons les valeurs manquantes des variables explicatives en apprentissage et en test par leurs moyennes respectives.

Afficher le code

Afficher la sortie masquée

Afficher le code

Afficher la sortie masquée

Régression linéaire

On lance une régression linéaire pour prédire Valeur foncière.

Afficher le code

```
Le RMSE est de : 100687.11018504541
```

Ce qui signifique qu'en moyenne, la régression linéaire prédit Valeur foncière avec une erreur de plus de 100 000 euros.

Decision Tree Regressor

On lance ensuite un Decision Tree Regressor afin de prédire Valeur foncière.

Afficher le code

```
Le RMSE est de : 97948.22407313711
```

Ce qui signifique qu'en moyenne, le Decision Tree Regressor prédit Valeur foncière avec une erreur de moins de 100 000 euros.

Random Forest Regressor

Nous lançons finalement un Random Forest Regressor.

Afficher le code

Le random forest était beaucoup trop long à lancer. Nous nous sommes donc contentés du Decision Tree Regressor, qui était notre meilleur modèle de prédiction, afin de prédire les valeurs de Valeur foncière.

Lien pour accéder au script sur Google Colab : https://colab.research.google.com/drive/1ov-zndGBtoySDJ_vmpuI74rJVIcc3RUy?usp=sharing