Google Colaboratory

1-Traitement des valeurs foncières

A - Importation des données

Première étape, on charge les 4 fichiers .txt correspondants aux données sur les valeurs foncières de 2018 à 2021.
On les regroupe en un seul dataframe 'df'.

'df' contient 15125102 lignes & 43 colonnes.

B - Nettoyage des données

On supprime les ventes qui apparaissent plus d'une fois le même jour ET à la même adresse:
-On créer un nouvelle colonne 'col_concat' avec l'adresse et la date de vente jointe.
-On supprime les lignes ayant la même valeur dans col_concat

On constate que plusieurs variables sont vides ou presques vide.
On supprime ces variables ainsi que certaines autres qui ne seront pas exploitées dans les modèles.

On supprime les lignes qui n'ont pas de 'type local' renseigné car c'est une variable utilisée pour l'entrainement des modèles.

On supprime les outliers concernant la valeur foncière par précaution pour ne pas perturber les modèles.

Étant donné que le but de nos modèle sera de prédire la valeurs foncière, correspondant donc à une vente, nous décidons de supprimer les lignes où 'Nature mutation' n'est pas 'vente'.

C - Open Data

On va maintenant enrichir le dataframe avec de l'Open Data.

Taux de chômage

On a trouvé un fichier sl_etc_2023T2.xls contenant le taux de chômage par département et par année.
Ça pourra nous aider dans les modèles de régression.
Après un nettoyage du fichier, nous avons pu rajouter à notre dataframe une nouvelle variable 'Moyenne Taux Chomage' indiquant la moyenne du taux de chômage
entre les années 2018 à 2021 par département.

Moyenne prix/m2/departement

On a créer deux nouvelles variables 'prix_par_m2' et 'moyenne_prix_par_m2_par_code_postal' calculée directement à partir des vraiables 'Valeur fonciere', 'Surface reelle bati' et 'Code postal' de notre dataframe. Ce seront des bons indicateurs pour nos modèles.

D - Exploration des données nettoyées

Voici quelques graphs permettant de se faire une idée du dataframe df

2 - Modèle de Classification sur la variable 'Type local'

Le but final du projet est de prédire la valeur foncière d'un bien. Une étape intermédiaire pour y parvenir est de créer un modèle pour prédire la variable 'type local', car c'est une variable explicative très importante qui nous servira ensuite pour les modèle de regression sur la valeur foncière.
Pour prédire le 'type local' nous utilisons un arbre de décision.

Création des groupes train et test

On va gerer les valeurs manquantes dans l'echantillon d'apprentissage

Apprentissage

Prediction

Optimisation des parametres de l'arbre

On va utiliser un greedsearch pour améliorer les performances du modèle

Les performances du modèle sont très bonnes.

3 - Reggresion sur la variable 'Valeur fonciere'

Maintenant que nous pouvons prédire la variable des 'Type local' nous pouvons passer à l'étape final qui est de trouver le meilleur modèle de régression afin de prédire les meilleures 'Valeur foncière'. Pour cette étape nous avons étudié pas mal d'options. Tout d'abord le choix des variables explicatives. Après pas mal de test, nous avons trouvé que les variables les plus pertinentes et qui renvoyaient les meilleurs résultats sont : 'Type local', 'Nombre pieces principales', 'Surface reelle bati', 'Surface terrain', 'Nombre de lots', 'Moyenne Taux Chomage', 'prix_par_m2', 'moyenne_prix_par_m2_par_code_postal'. A cela nous avons ajouté des données OpenData qui sont le prix par m2 de la vente et la moyenne du prix par m2 dans le département.

A - Echantillonage

Pour les données manquantes, nous les avons remplacés par la moyenne de la variable. On a aussi dû traiter que des variables étaient 'inf' (venant de notre calcul du prix par m2). Ces valeurs infinies ont été remplacée par la moyenne du prix au m2 dans son département.

Pour ne pas donner plus d'importance aux variables explicatives à forte variance, il est essentiel de centrer et réduire les données en amont. On centre et réduit également afin de les ramener à la même échelle.

B - Apprentissage

Pour l'apprentissage de notre modèle nous en avons testé des différents : Régression linéaire multiple, Ridge, Lasso, ElasticNet, Arbre de décision ou encore RandomForest.

Pour maximiser les performances de nos modèles, nous avons effectué des GridSearch sur chacun d'eux afin de récupérer les meilleurs hyper-paramètres possibles. Lasso, ElasticNet et RandomForest

prenaient un temps considérable d'éxecution (plus de 2h), et nous avons donc pas testé jusqu'au bout, mais lors de test manuel nous pouvions voir qu'ils ne seraient pas les plus satisfaisant. Tout comme la régression linéaire qui n'a pas rendu des très bon scores. Ci-dessous je vous montre 2 exemples de modèles qui nous ont parus performant : **Ridge** et **Arbre de Décision**.

Régression Ridge

On peut noter ici que les variables les plus significatives sont 'Nombre pieces principales' et 'moyenne_prix_par_m2_par_code_postal', tandis que le 'prix_par_m2' et le 'type_local' ont leur importance aussi dans le modèle.

Arbre de décision

Les résultats de ces deux modèles furent les plus satisafaisants. Nous sommes donc partis sur la régression avec l'arbre de décision pour la prédiction des valeurs foncières sur le jeu de données Kaggle.

4 - Interface Dash

Pour l'interface Dash, nous avons décider de faire 3 onglets. Un onglet statistiques avec les graphiques que nous avons effectués précédement. Un onglet Cartographie, pour visualisre nos données sur le territoire Français. Et un onglet Prédictions pour prédire/ Estimer un bien immobilier. L'interface demande un certain temps de chargement des données, c'est pour cela que nous avons limités les donnnées à 8000 lignes. Avant cela, nous avons évidemnt mélanger la datframe d'origine.

A - Statistiques

Pour les statisques, nous avons récupérés les graphiques pour étudier nos données. Deux Filtres ont été créés sur cette page:

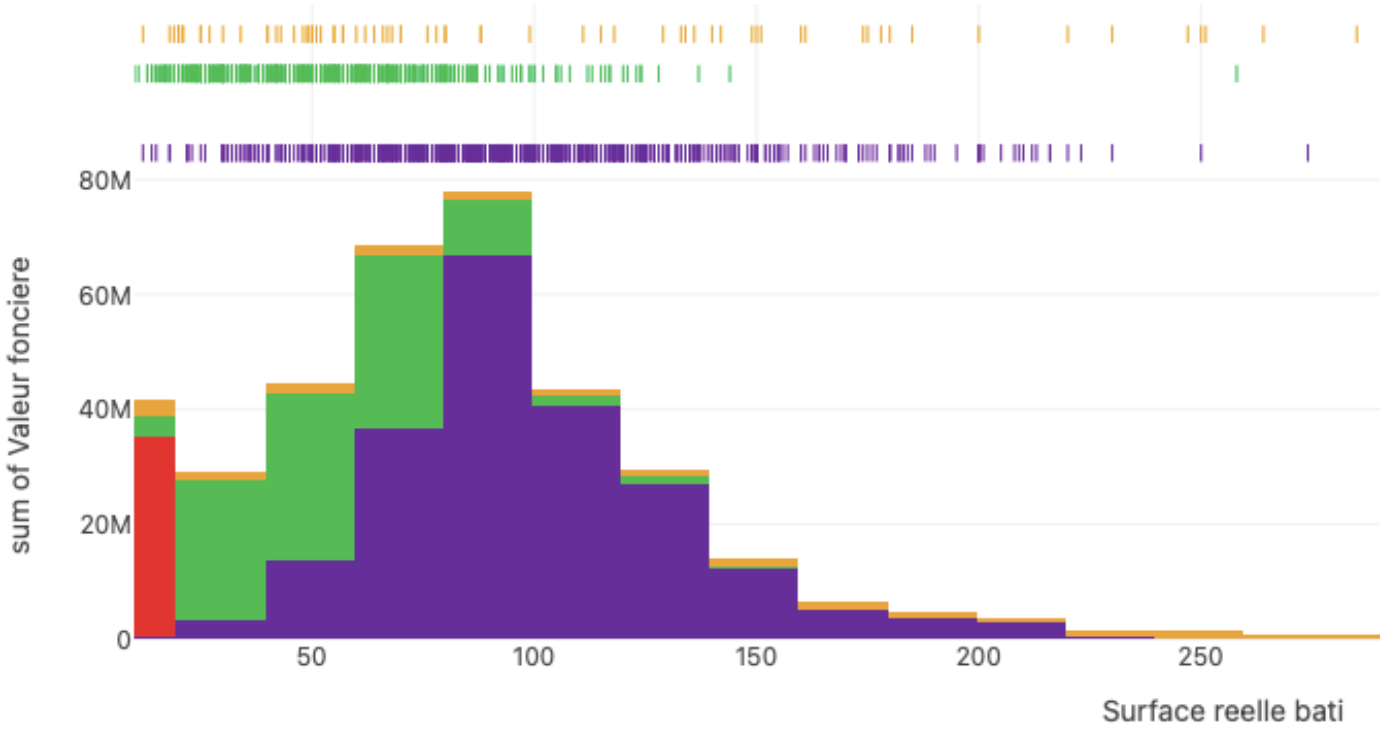
- un filtre sur l'année des données
- un filtre sur le type local des donnés

Ces filtres sont reliés aux graphiques et sont modifiés une fois les filtres selectionnés.

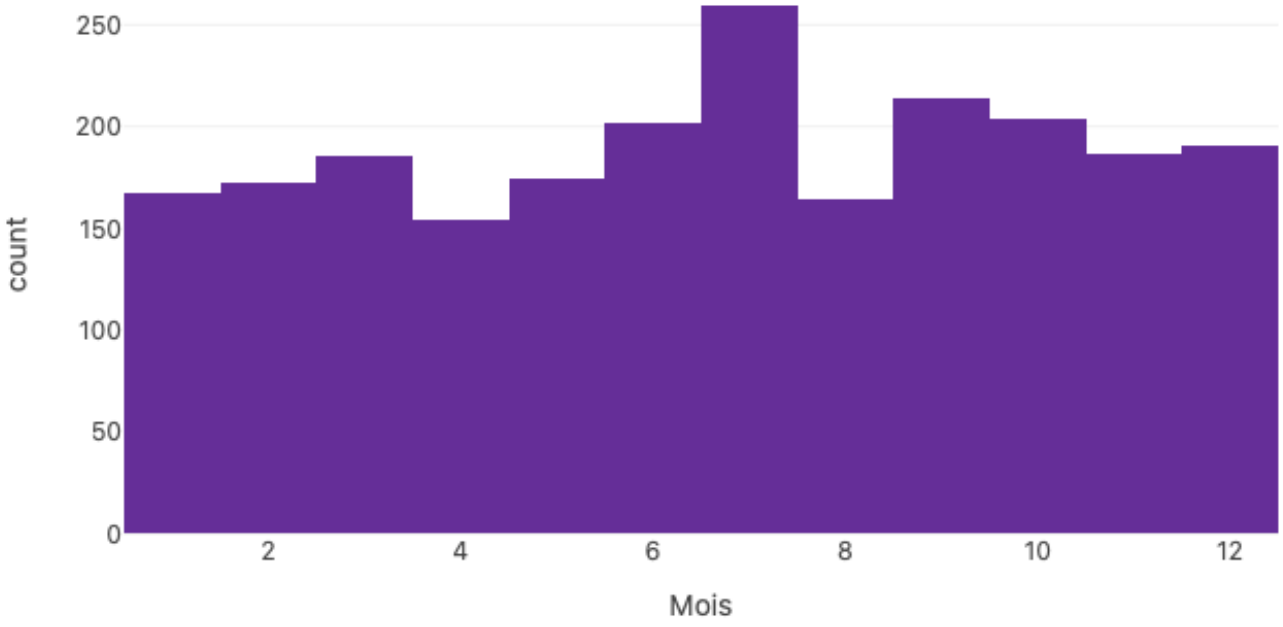
Tous les graphiques sont exportables sous format png à l'aide de plotly.express. Comme on peux le voir sur la photo ci dessous, le graphique en bas à droite est exportable avec le pictogramme de plotly.express.


Statistiques
Sta
Sélectionnez une année :
ALL
Sélectionnez un
2018
2019
2020
2021
ALL
Moyenne des prix des
LL
avg of Valeur fonciere
150k
100k
50k
0
Maison
Appartement
Dépendance
Local industriel. commercial ou assimilé
Local industriel. commercial ou assimil
Type local

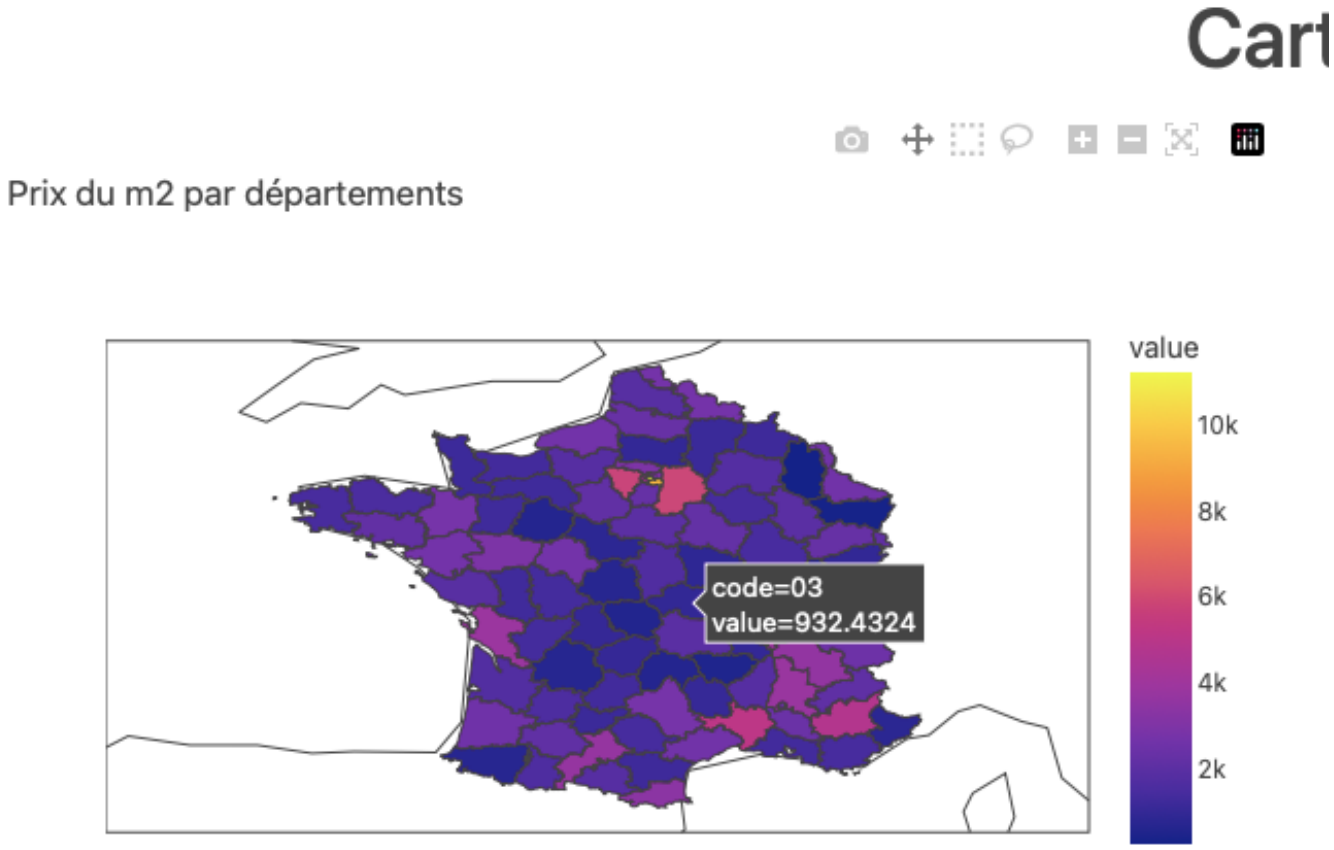
image.png

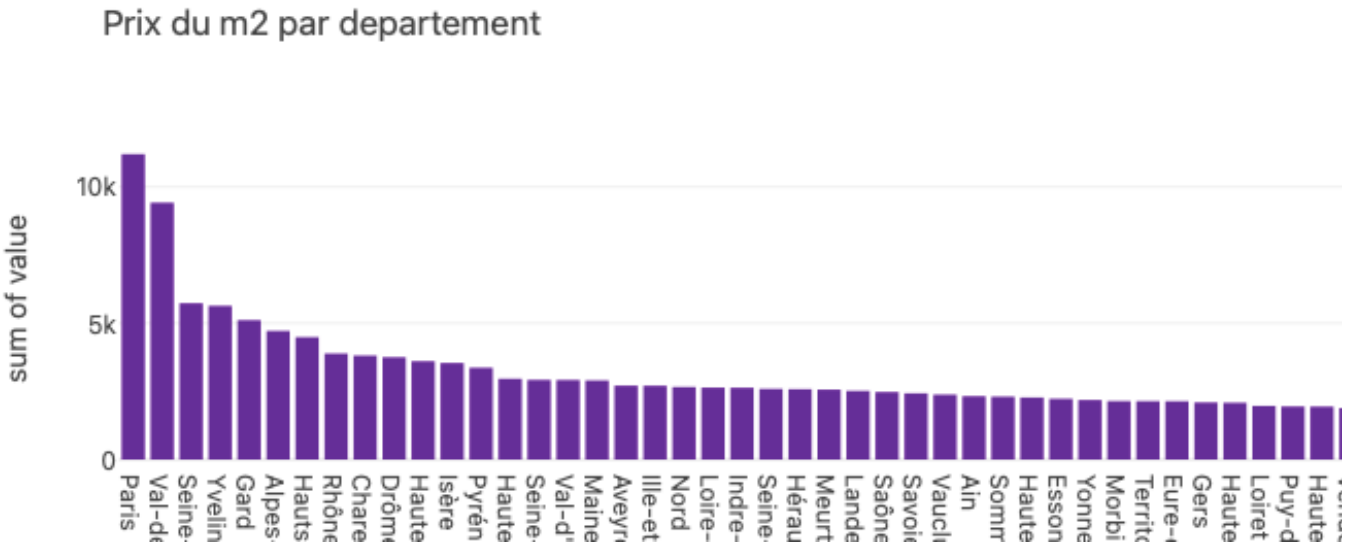


Nombre de Ventes par mois en 2019 pour les biens ALL



B - Cartographie





La Cartographie a été effectuée dans un premier temps sur les données initiales pour afficher le prix du mètre carré par département. Chaque Bien immobilier a eu son prix au mètres carré puis ce prix a été regroupé par département avec une moyenne.

Ensuite, pour notre apprentissage des données Open Data sur le taux de chômage par départements a été ajouté. Il nous semblais pertinent d'effectuer une cratographie sur ce taux de chômage.

Un histogramme a été également pertinent pour mieux voir quels sont les départements avec les prix au mètres carré le plus/le moins cher.

C - Prédictions

Pour la partie des prédictions, un bien est estimé à partir de plusieurs champs que l'utilisateur peut remplir. Si l'utilisateur ne remplit pas ou ne connait pas le type local de son bien, nous utilisons la classification que nous avons mis en place afin de prédire le type du bien.

Par la suite le prix du bien est prédit à l'aide du meilleur modèle de régression que nous avons effectué.

Des règles sur ces champs sont évidements présentes. Si la surface réelle bati n'est pas rempli par exemple, nous demandons à l'utilisateur de remplir correctement les champs.

Logo Markdown