

PROJET MACHINE LEARNING SUPERVISE

SeLoger

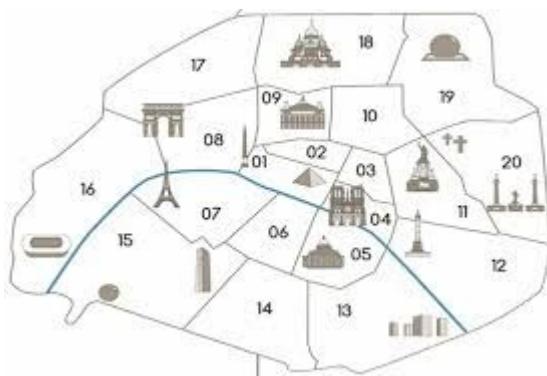
Introduction

Sujet: Régression (prédiction prix)

Site : Se loger.com

En tant qu'agent immobilier, le but est de conseiller les particuliers sur le prix de la vente de leur appartement/maison (tout en justifiant vos choix).

PARIS



LYON

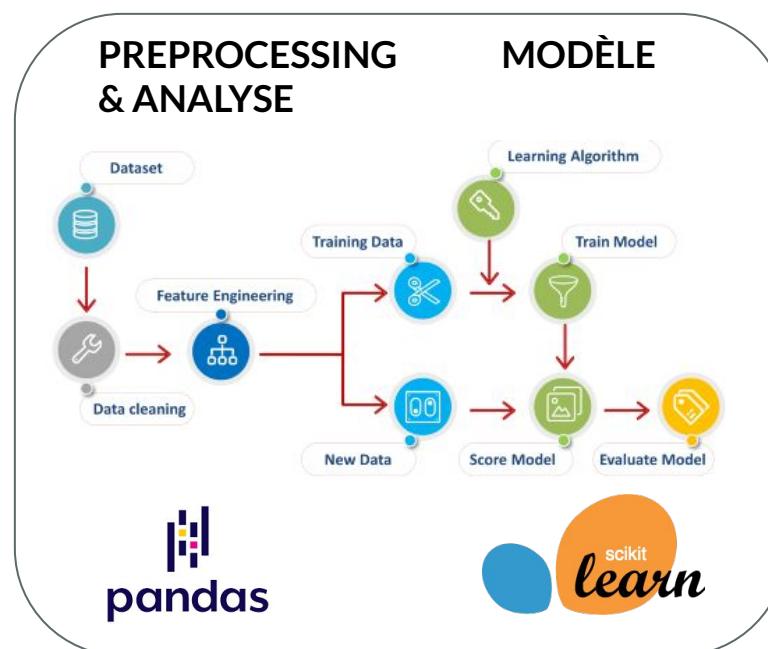


Plan

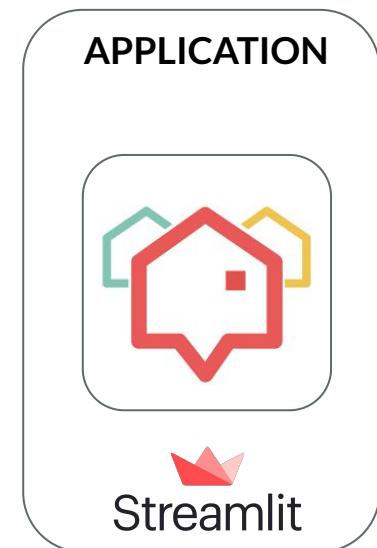
PARTIE 1



PARTIE 2 & 3



PARTIE 4



Etape 1 : Acquérir les données (Scraping)

Sélection de la liste des critères de recherches afin d'avoir plus d'informations :

- Le lieu : Paris 16 ème et Lyon 6ème
- Type de bien : Appartement et Maison
- Type de logement : Ancien

Identification des informations à scrapper

Collecter les données nécessaires permettant de prédire le prix final de chaque maison : **29 catégories**

- Type de logement, Nombre de pièce, Nombre de chambre, Mètre carré, Arrondissement, Quartier , Adresse, Prix de d'achat du bien en euros, Nombre toilette, Année de construction
- Informations pour chaque logements choisis : Récupérer le détails de descriptions des logements pour chaque catégories suivantes (si existante): Cave, Jardin, Parking, piscine, gardien, balcon, terrasse, étage, vue, Hauteur, orientation, ascenseur, parquet.

Photos Visite virtuelle 3D Quartier



Sauvegarder

Appartement

Quartier Auteuil Sud, Paris 16ème

4 pièces 3 chambres 77m² 10 338 € / m²

796 000 €

À partir de 3 289 € / mois



A l'intérieur

1 Salle de bain
Toilette séparée
Cuisine séparée
Salle de séjour

Autres

Travaux
Digicode

[Afficher moins ▾](#)

1 Toilette
Chauffage gaz collectif radiateur
Une entrée

Interphone
Orientation Est

Appartement

Quartier Auteuil Sud, Paris 16ème

796 000 €

À partir de 3 289 € / mois



À propos de cet appartement 4 pièces à Paris 16ème

L'avis du professionnel

Michel Ange - Auteuil VENDMY, première néo-agence internationale à commission réduite vous présente:
Situé à deux pas de tous commerces et transports, au 4ème étage avec ascenseur d'un très bel immeuble 1930 bien entretenu, gardien et sécurisé, appartement très bien configuré, à refaire, traversant, avec vue

[Afficher plus ▾](#)

Les plus



Cave Vue Gardien Ascenseur

Général

Surface de 77m²

4 Pièces

3 Chambres

Année de construction 1930

Au 4ème étage

[Afficher plus ▾](#)

BELLES DEMEURES

Finest properties only.

< Retour à votre recherche (191)

Les plus



Terrasse



Ascenseur



Belle vue

Caractéristiques générales

Dernier étage

Au 3ème étage

Année de construction 1900

5 Pièces

Bâtiment de 3 étages

Aménagement du bien

Salle de séjour

1 Chambre

Parquet

Salle à manger

Non meublé

1 Salle de bain

Entrée

Cheminée

Diagnostic de performance énergétique

DPE non communiqué

BELLES
DEMEURES
Finest properties only.

ACHETER LOUER VACANCES RÉGIONS MAGAZINE LES AGENCE

< Retour à votre recherche (191)

< Propriété 5/191 >

Heart Mettre en favoris



Aidez-nous à améliorer le site Belles Demeures

ACCUEIL > FRANCE > PARIS > QUARTIER CHAILLOT > VENTE APPARTEMENT LUXE > VENTE APPARTEMENTS 5 PIÈCES 230 M²

Découvrir des propriétés similaires

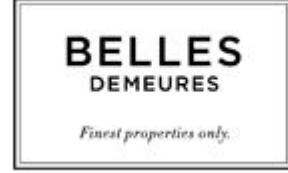
Les actualités du luxe

Lieux exclusifs, événements prestigieux
restez informé du marché du luxe.

Votre email

OK

Volumétrie :

	<i>SeLoger</i>		
PARIS (16e)	436	239	675
LYON (6e)	201	38	239
TOTAL	637 annonces	277 annonces	914

Les difficultés rencontrées



We are sorry...

...but your activity and behavior on this website made us think that you are a bot.

1. To protect this website, we cannot process your request right now.
2. If you think this is an error, please [contact us](#) copying below ID in the email.

Please solve this CAPTCHA in helping us understand your behavior to grant access

I'm not a robot 
reCAPTCHA
[Privacy](#) • [Terms](#)

SeLoger

Il semble que vous êtes nombreux à vous connecter depuis ce réseau ...

Merci de nous confirmer que vous n'êtes pas un robot :

I'm not a robot 
reCAPTCHA
[Privacy](#) • [Terms](#)



Comment un site peut détecter le scrapping ?

1. **Trafic important, spécialement provenant d'un client ou adresse IP unique** dans une courte période de temps
2. **Tâche effectuée sur le site répétitive** - base sur le fait qu'un humain ne fera pas toujours la même tâche d'une manière si répétitive
3. **Website : Changing Layouts**

<https://www.scrapehero.com/how-to-prevent-getting-blacklisted-while-scraping/>



Afin d'éviter d'être bloqué, on peut demander à notre robot de se présenter correctement, comme un humain, en précisant un bon User_agent.



Enfin, à chaque passage, nous pouvons changer l'adresse IP du robot.

Proxy: joue le rôle de passerelle entre Internet et vous

Voici le motif classique d'un user-agent :

Mozilla/[version] ([system and browser information]) [platform] ([platform details]) [extensions]

Les parties soulignées et en couleur sont variables.

Liste des annonces



URL des annonces



MAIS PARFOIS:

~~URL
SeLoger~~

**URL
Belle
Demeure**

ET: Changing Layouts

Etape 2 : DataFrame

2.1 Découverte environnement DataFrame

- **Création de 2 dataframes pour chaque site** (SeLoger et Belle Demeures) pour Paris ou Lyon.
- **Concaténer les 2 dataframes pour avoir un DataFrame unique** correspondant à la ville (Paris ou Lyon)

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 675 entries, 0 to 238
Data columns (total 29 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   sl_style          637 non-null    object  
 1   sl_localisation   533 non-null    object  
 2   sl_nb_chambre     567 non-null    float64 
 3   sl_taille         624 non-null    float64 
 4   sl_prix           622 non-null    float64
```

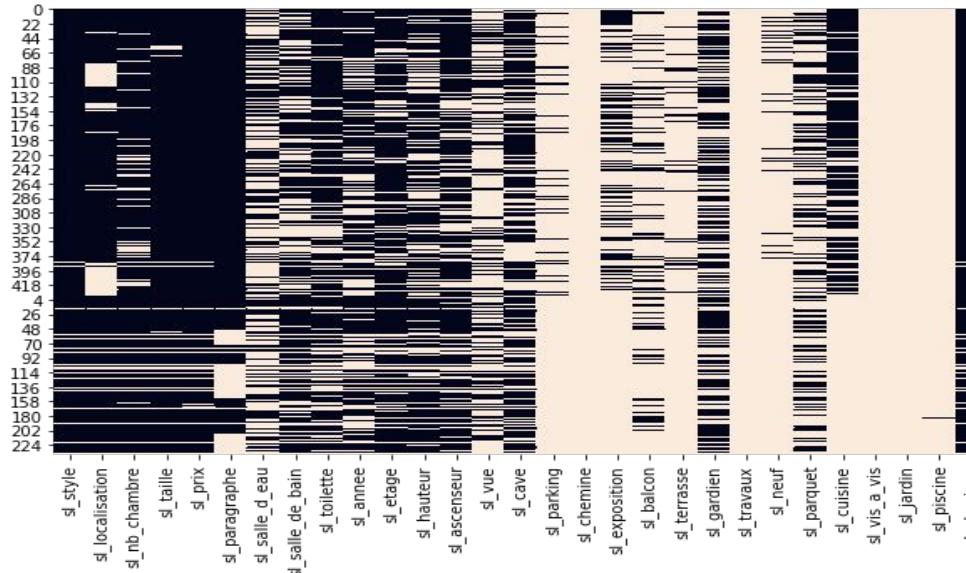
- Pour Paris : 675 observations au total(436 observations avec SeLoger et 239 observations avec Belle Demeure) et 29 variables (15 variables continues et 14 variables qualitatives).
- Pour Lyon: 239 observations au total (201 observations avec SeLoger et 38 observations avec Belle Demeure) et 29 variables (15 variables continues et 14 variables qualitatives).

Etape 2 : DataFrame

2.2 Visualiser les valeurs nulles

Dans le graphique ci-dessous, on remarque que les valeurs non nulles sont supérieures aux valeurs nulles. Certaines colonnes contiennent que des “NaN” et devront être supprimées ("sl_jardin", "sl_vis_a_vis", etc).

```
def viz_na(data):
    return sns.heatmap(data.isna(), cbar=False)
```



Etape 3 : Analyse des données

3.1 Analyse des données Target

```
data.sl_prix.describe()
```

```
count      6.330000e+02
mean       1.927855e+06
std        2.464604e+06
min        5.500000e+04
25%        6.900000e+05
50%        1.150000e+06
75%        2.420000e+06
max        3.900000e+07
Name: sl_prix, dtype: float64
```

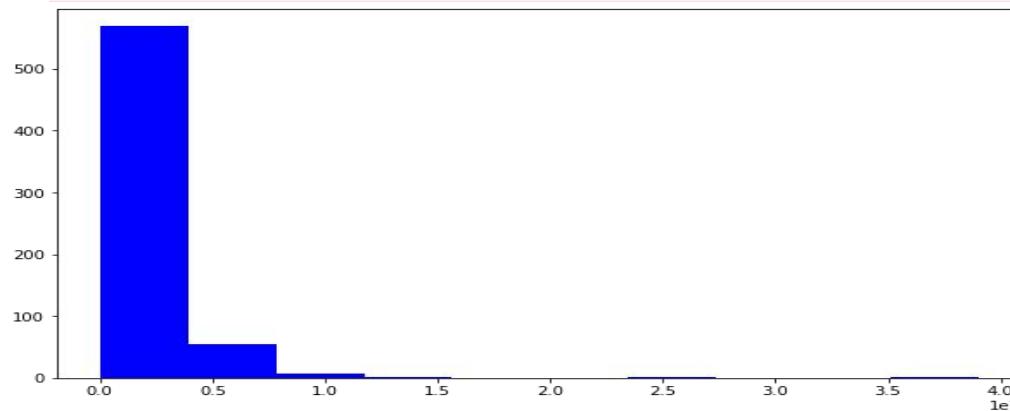
Pour Paris, on remarque que **le prix moyen du logement est proche de 1 928 000€**, avec la plupart des **valeurs comprises entre 690 000€ et 2 420 000 €**

Etape 3 : Analyse des données

3.1 Analyse des données Target

```
plt.hist(data.sl_prix, color = 'blue')
print ("skew=", data.sl_prix.skew())
```

skew= 7.367181486092676



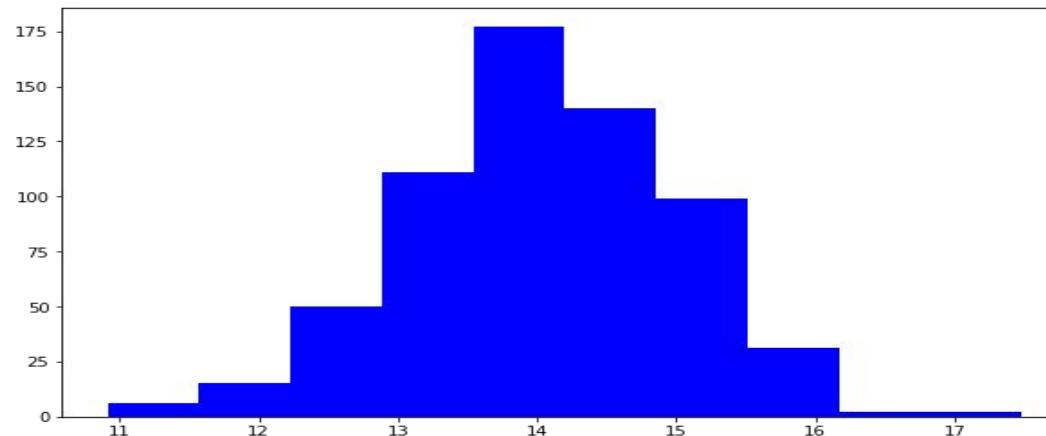
Dans le graphique ci-dessus, on constate que **la distribution est asymétrique vers la droite**.

Etape 3 : Analyse des données

3.1 Analyse des données Target

Pour améliorer la linéarité des données, on applique la fonction `np.log()` sur le target. Nous pouvons voir visuellement que les données ressemblent davantage à une **distribution normale**.

```
target = np.log(data.sl_prix)
plt.hist(target, color = 'blue')
print ("skew=", target.skew())
```



N.B : `np.exp()` permet d'inverser la transformation.

Etape 3 : Analyse des données

3.2 Relation entre target et features numériques

Les 2 premières features sont les plus positivement corrélées avec le prix, tandis que les 2 derniers sont les plus négativement corrélées.

```
numeric_features_p = data.select_dtypes(include=[np.number]).corr()  
  
print ("Correlation Paris", '\n', numeric_features_p['sl_prix'].sort_values(ascending=False)[:5], '\n')  
print (numeric_features_p['sl_prix'].sort_values(ascending=False)[-5:])
```

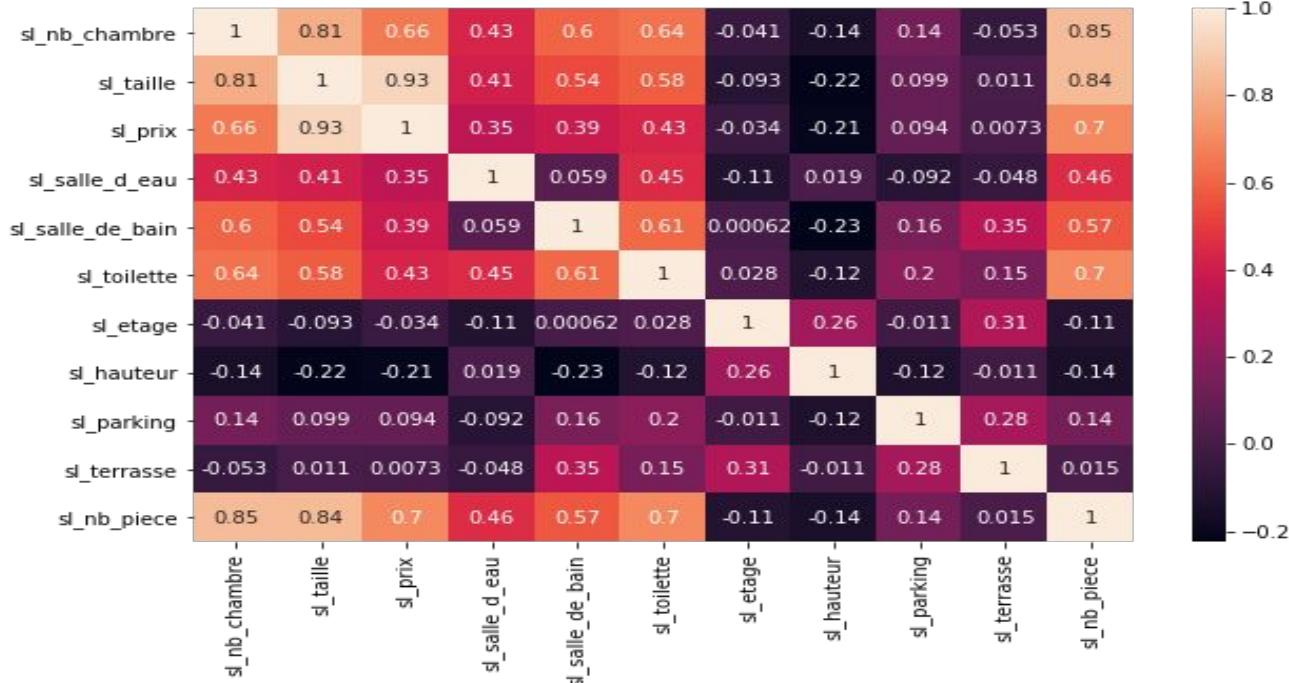
```
Correlation Paris  
    sl_prix      1.000000  
    sl_taille    0.928970  
    sl_nb_piece  0.701882  
    sl_nb_chambre 0.657720  
    sl_toilette  0.434425  
Name: sl_prix, dtype: float64
```

```
    sl_salle_d_eau   0.350353  
    sl_parking      0.094498  
    sl_terrasse     0.007312  
    sl_etage        -0.034272  
    sl_hauteur      -0.207661  
Name: sl_prix, dtype: float64
```

Etape 3 : Analyse des données

3.2 Relation entre target et features numériques

```
sns.heatmap(numeric_features_p, annot=True)
```



Etape 3 : Analyse des données

3.3 Relation entre target et features non numériques

```
categoricals = data.select_dtypes(exclude= [np.number])
categoricals.describe()
```

	sl_style	sl_localisation	sl_annee	sl_ascenseur	sl_vue	sl_cave	sl_exposition	sl_balcon	sl_gardien	sl_neuf	sl_parquet	sl_cuisine
count	637	533	435	505	223	442	173	178	382	52	259	343
unique	4	8	85	1	2	20	8	26	1	1	1	8
top	Appartement	Paris 16ème	Année de construction 1900	Ascenseur	Vue	Cave	Orientation Sud	1 Balcon	Gardien	Refait à neuf	Parquet	Cuisine séparée
freq	613	206	100	505	133	405	45	73	382	52	259	141

count : indique le nombre d'observations non nulles.

unique : indique le nombre de valeurs uniques.

top : indique la valeur la plus courante.

freq: la fréquence de la valeur maximale indiquée

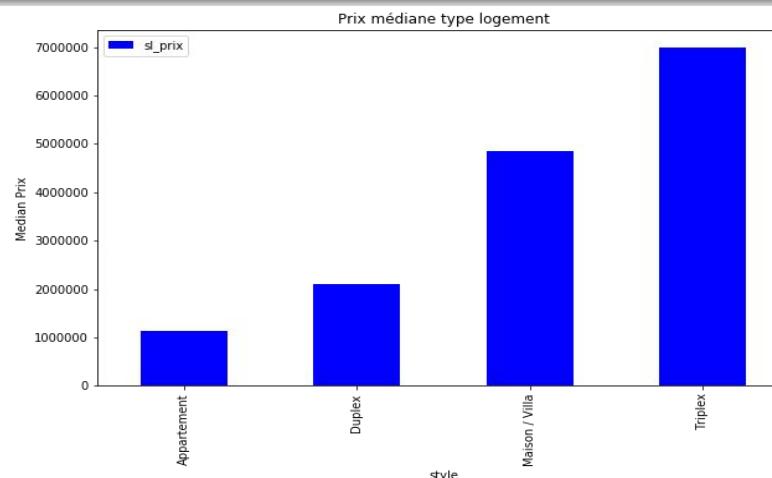
Etape 3 : Analyse des données

3.3 Relation entre target et features non numériques

Notons que le prix d'achat médiane d'un logement de type Triplex est plus élevé que les autres.

```
▶ cuisine_pivot = data.pivot_table(index='sl_style', values = 'sl_prix', aggfunc = np.median)
print(cuisine_pivot)

cuisine_pivot.plot(kind='bar', color='blue')
plt.xlabel('style')
plt.ylabel('Median Prix')
plt.title('Prix médiane type logement')
```



Etape 4 : Encoding Categorical Features

Comme vu précédemment, nous constatons que le **prix d'achat médiane de la Cuisine séparée équipée, Triplex, et le Paris 16ème sont nettement plus élevés que les autres**. Pour se faire nous créons des nouvelles colonnes.

```
def encode(x):
    return 1 if x =='Triplex' else 0

data['sl_style_enc'] = data.sl_style.apply(encode)

df.sl_localisation.value_counts()
Quartier Les Brotteaux-Bellecombe-Masséna      61
Quartier Tête d'Or-Foch-Vitton                  54
Name: sl_localisation, dtype: int64

def encode(x):
    return 1 if x =='Paris 16ème ' else 0

data['sl_localisation_enc'] = data.sl_localisation.apply(encode)

def encode(x):
    return 1 if x =='Cuisine séparée équipée' else 0

data['sl_cuisine_enc'] = data.sl_cuisine.apply(encode)
```

Etape 4 : Encoding Categorical Features

Nous choisissons 1 pour la value_counts() la plus courante et 0 dans l'autre cas : **création de nouvelles colonnes**

```
def encode(x):
    return 1 if x == 'Refait à neuf' else 0
data['sl_neuf_enc'] = data.sl_neuf.apply(encode)

def encode(x):
    return 1 if x == 'Cave' else 0
data['sl_cave_enc'] = data.sl_cave.apply(encode)

#data.sl_annee.value_counts()

def encode(x):
    return 1 if x == 'Année de construction 1900' else 0
data['sl_annee_enc'] = data.sl_annee.apply(encode)

#data.sl_balcon.value_counts()

def encode(x):
    return 1 if x == '1 Balcon' else 0
data['sl_balcon_enc'] = data.sl_balcon.apply(encode)

def encode(x):
    return 1 if x == 'Parquet' else 0
data['sl_parquet_enc'] = data.sl_parquet.apply(encode)

def encode(x):
    return 1 if x == 'Gardien' else 0
data['sl_gardien_enc'] = data.sl_gardien.apply(encode)
```

Etape 5 : Data Final

5.1 Missing data : Interpolation

Cette méthode permet de remplir les valeurs manquantes par une valeur moyenne.

```
data_paris = data.select_dtypes(include=[np.number]).interpolate().dropna()  
data_paris
```

	sl_nb_chambre	sl_taille	sl_prix	sl_salle_d_eau	sl_salle_de_bain	sl_toilette	sl_etage	sl_hauteur	sl_parking	sl_terrasse	...	sl_exposition_enc	sl_exposition
8	3.0	217.0	1995000.0	1.5	3.0	1.0	4.0	6.0	2.000000	3.000000	...	0	0
9	3.0	156.0	2950000.0	1.0	1.0	1.0	2.0	6.5	1.933333	2.333333	...	0	0
10	1.0	55.0	595000.0	1.0	1.0	1.0	4.0	7.0	1.866667	1.666667	...	0	0
11	1.0	25.0	370000.0	1.0	1.0	1.0	4.0	7.0	1.800000	1.000000	...	0	0
12	1.0	44.0	455000.0	1.0	1.0	1.0	5.0	7.0	1.733333	1.000000	...	0	0
...
234	6.0	293.0	4672500.0	2.0	2.0	1.0	5.0	7.0	2.000000	1.000000	...	0	0
235	3.0	74.0	915000.0	2.0	1.0	1.0	5.0	5.0	2.000000	1.000000	...	0	0
236	6.0	425.0	6825000.0	2.0	3.0	2.0	1.0	6.0	2.000000	1.000000	...	0	0
237	1.0	45.0	686000.0	2.0	3.0	2.0	3.0	6.0	2.000000	1.000000	...	0	0
238	1.0	45.0	686000.0	2.0	3.0	2.0	3.0	6.0	2.000000	1.000000	...	0	0

667 rows × 23 columns

Etape 6 : Modèle

6.1 Choix du Model - Modèle Linéaire

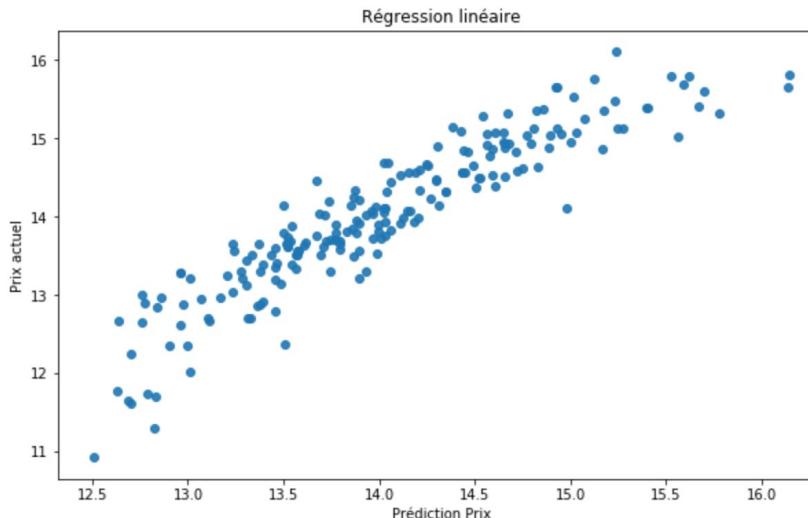
RMSE :

Le RMSE est proche de 0.

La **distance entre les valeurs réelles et les valeurs prévues est quasi nulle.**

```
| from sklearn.metrics import mean_squared_error
| def RMSE (y_test, y_pred):
|     return mean_squared_error(y_test, y_pred)
| # La distance entre nos valeurs prévues et les valeurs réelles.
| RMSE(y_test, y_pred)
:
: 0.16638479405987075
```

```
| #prediction_f = np.exp(y_pred)
| #y_test = np.exp(y_test)
| plt.scatter(y_pred, y_test, alpha=0.9)
| plt.xlabel('Prédiction Prix')
| plt.ylabel('Prix actuel')
| plt.title('Régression linéaire')
| plt.show()
```



Etape 6 : Modèle

6.2 Choix du Modèle - Bilan

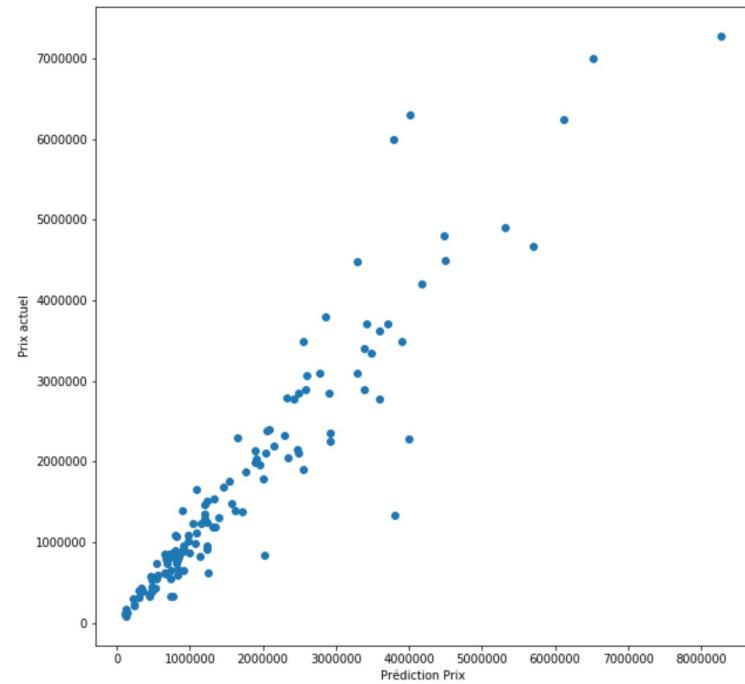
Modèle pour PARIS	LR	Lasso	Ridge	Elastic net	GradientBoosting Regressor
R2 (test)	0.812395		0.808257		0.930533
R2 (train)	0.825019		0.814815		0.999877
CV Score	0.738828	0.738959	0.740621	0.740579	0.927142
RMSE	0.166384		0.169880		0.053219

D'après ce tableau, le **modèle Elastic_net semble être le plus robuste** pour prédire le prix. D'autre part, on remarque que le Rmse de GradientBoostingRegressor est le plus faible.

Etape 7 : Prédiction, Prix avec GradientBoostingRegression

```
predict_prices = clf.predict(X_test)
predict_final = np.exp(predict_prices)
y_test = np.exp(y_test)
```

```
plt.scatter(predict_final, y_test, alpha=1)
plt.xlabel('Prédiction Prix')
plt.ylabel('Prix actuel')
plt.title('Régression linéaire')
plt.show()
```



Etape 8 : Application - DEMO

✓ Prediction du prix de votre logement

Faites varier les features pour connaître le meilleur prix de vente de votre appartement ou maison

Ville

Localisation du logement

Paris

Lyon

Quartier

Localisation du logement

Paris 16

Autres arrondissements

Type du logement

Style

Appartement

Combien de mètre carré

PredictLogement

L'application qui vous aide à prédire le prix de votre logement au bon prix - par SeLoger.com



Points d'amélioration

- Scrapping à l'instant donné du modèle : réactualisation
- Besoin d'une collecte de données approfondies
- Prix parfois pas réaliste - révision du modèle
- Problème de restriction entre étage et bâtiment
- Essayer de faire la prédiction du prix dans un mois/un an (afin de voir si le prix est inférieur ou supérieur au prix à l'instant)